INTERFACE AGE ™

COMPUTING FOR HOME AND BUSINESS APPLICATIONS   VOLUME 3, ISSUE 10 OCTOBER 1978 $2.00
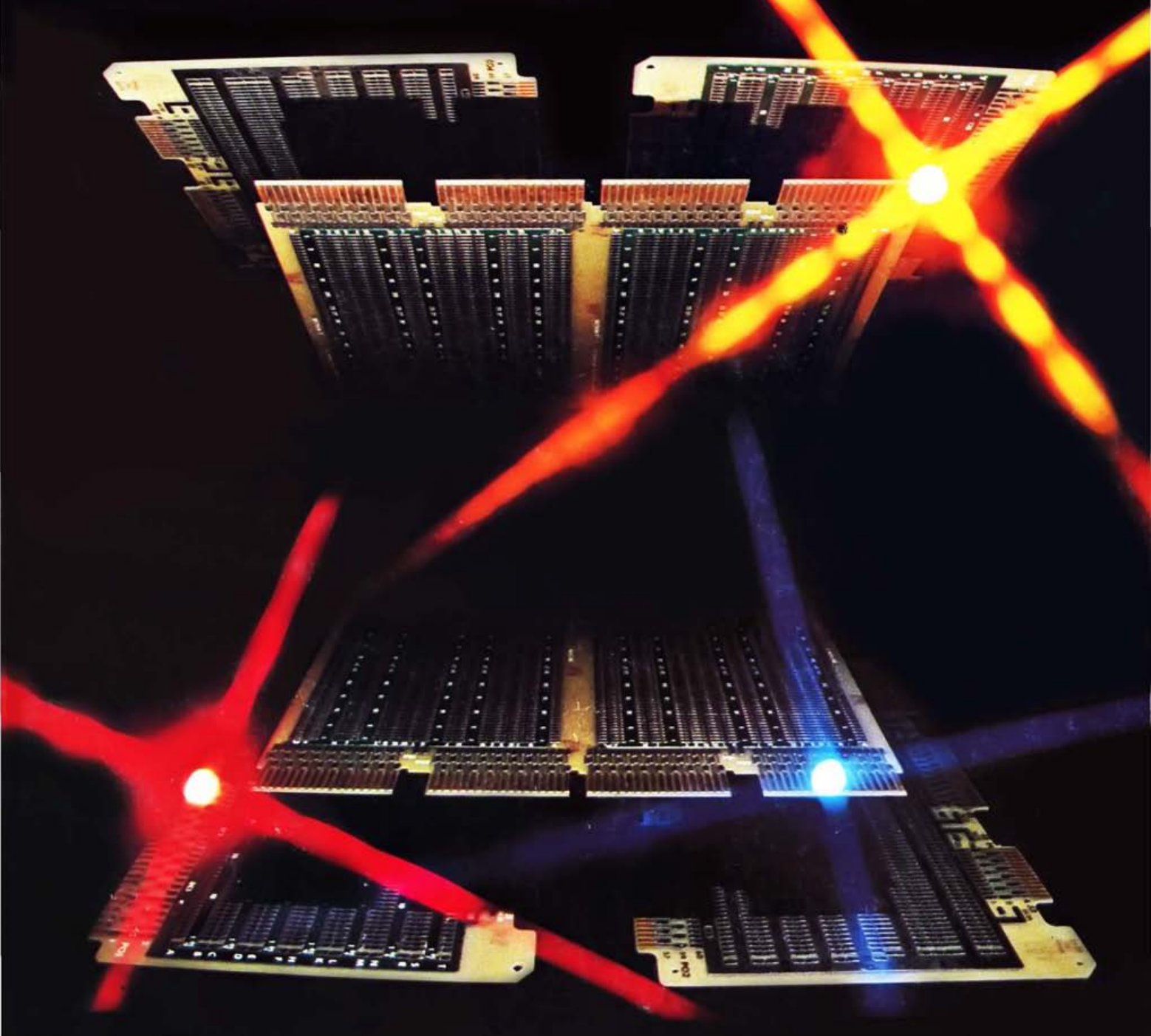
CANADA/MEXICO $2.50

INDEX TO HARDWARE

Microprocessors in Autos

The 6800 in the Darkroom

INTERFACE AGE ™

COMPUTING FOR HOME AND BUSINESS APPLICATIONS

INDEX TO HARDWARE
Microprocessors in Autos

THIS MONTH'S COVER

Hardware has been the primary concern of the new microcomputer industry and in the last three years, the capabilities have become almost unbelievable in respect to the original designs.

The cover depicts the printed circuit boards that carry the data signals within the computer system. The reflective convergence of the cover epitomizes the industry's goal to bring dreams to reality.

The cover was provided courtesy of Multi Link, 2121 South Manchester Avenue, Anaheim, California, and is an example of their custom designed planar boards.

EDITOR'S NOTEBOOK

The last several weeks have really been exciting from my viewpoint. Many manufacturers are finishing up their new entrants to the market and are really starting to get enthused.

Nancy Millican of MECA, the people who make the Alpha I tape system, informed me that they are about to announce their Delta I disk system. The Delta I works in tandem with their tape system and provides the very first total storage package for the microcomputer market. Nancy said that with the way things are going it looks like they wil be able to ship by late November or at the latest early December.

Another company that has been doing some really exciting work is "exatron" of Sunnyvale, California. This little company owned by Bob Howell is the manufacturer of a unique device called the "Stringy Floppy". This machine is a small continuous tape loop device which provides the speed of a floppy and the convenience of tape. The Stringy Floppy is designed primarily for S-100 bus type machines, but Bob has been working the last several months on developing an interface for 6800 based machines. From last reports it's just around the corner and will appear first in the pages of INTERFACE AGE.

A LOOK AT SMOKE SIGNAL BROADCASTING

This month's profile is about a small company known to 6800 users called Smoke Signal Broadcasting. This company came into being about a year and a half ago with the sole purpose of supplying extras to the Southwest Technical Products 6800 microcomputer.

Originaly Smoke Signal developed a 16K static RAM board which works extremely well in the SWTP system. However, in the last several months they have been developing 5.25-inch floppies with controller and are currently planning an 8-inch disk system.

With the Floppy system they provide a disk operating system that makes use of random access files. We plan a review by Bill Turner on the capabilities of this DOS.

Smoke Signal is in the process of preparing their very first total computer system, as of this writing still unnamed. This 6800 system utilizes a well designed mother board with gold molex pin connectors, built-in dual floppies, and up to 64K of memory. The formal an-

nouncement of the system will be in the November New Products directory of INTERFACE AGE.

The guiding light behind Smoke Signal is Ric Hammond, president. Ric has taken the approach that the way to run a microcomputer business is to stay current with the market needs but at the same time avoid trying to market too many products at a time.

Apparently this philosophy has worked well for them. As a result Smoke Signal has not suffered the financial difficulties that are plaguing several other manufacturers.

Heading up the software side of Smoke Signal is Roger Embree, manager of software systems. Roger developed the DOS and has made decisions to use the TSC assembler/editor and Software Dynamics BASIC compiler as part of the supplied software packages. They also supply a BASIC interpreter from Computerware. Roger's goal is to ensure that only the best possible software is supplied to Smoke Signal users.

Rounding out the team is Ed Martin, marketing director. Ed brought over 14 years of hardware and marketing knowhow to the company. When I asked him why he left the security of a larger company to come to Smoke Signal, his reply was like most of ours in this industry: "I see a dynamic company with a great big chance to grow and a place where my marketing talents can be really put to the test."

Ed is obviously up to the test; through his efforts Smoke Signal has really become one of the major contenders for the 6800 market share.

Smoke Signal is a small growing company with a great outlook. For those users and distributors interested in more information regarding Smoke Signal and their products contact: Ed Martin — Marketing Director, Smoke Signal Broadcasting, 6394 Yucca, Hollywood, CA 90028, or call (213) 462-5652.

ANOTHER IMPORTANT BOOK AVAILABLE

Books are one of the important tools of our industry, but it is sometimes difficult to find the exact one you need to help solve a problem. The folks at The Computer Bookstore recently sent me a pre-release copy of the booklet: *Master List of Computer Books,* spring edition. This booklet sells for $1.50 and is really a must to round out the computer reference library. The book lists a number of reference sources and provides an order form so a reader can order them quickly. The booklet can be obtained from The Computer Bookstore, 796 Navy Street, Fort Walton Beach, FL 32548.

MYKRO CORPORATION AND IAPS™

George Sutton, the president of Mykro Corporation just recently sent me the updated version of the IAPS™

Is your TRASH FLOW higher than your CASH FLOW?

SOFTWARE LIBRARY

Five diskettes are included to give you immediate operating and programming capabilities.

DISK 1 is a master diskette with BASIC, MDOS, Text Editor, Assembler and more!

DISK 2 has many games including LUNAR, CRAPS, and SPLAT. There's room left for you to add your own.

DISK 3 contains a Small Business Accounting package which includes Accounts Payable and Receivable, Inventory, a General Ledger, and more.

DISKS 4 & 5 are blank so you may add your own programs.

Clean it up with a VERSATILE 3B™

If you're a businessman, we know how difficult it is to keep neat and efficient accounting records. Let our **VERSATILE** systems do it for you. You'll have a complete system built into a single cabinet, and a free software library on diskette to get your computer working for you the first day. $3295 Assembled and tested.

WHAT ELSE DOES A COMPLETE SYSTEM NEED?

Contact Our Distributors for Regional Sales and Service

Alexander & Co., Inc.
5518 Florin Road
Sacramento, CA 95823
(916) 422-9070

Huron Electronics
415 N. Silver
Bad Axe, Mich. 48413
(517) 269-9267

The Computer Store
3801 Kirby Dr., Suite 432
Houston, Texas 77098
(713) 522-7845

Southeast Representative:

Scientific Sales Co.
175 W. Wieuca, Suite 210
Atlanta, GA 30342
(404) 252-6808

Dealer Applications Available.

COMPUTER DATA SYSTEMS

5460 Fairmont Drive • Wilmington, Delaware 19808 • 302-738-0933

DYNABYTE COMPUTERS ARE ALL BUSINESS INSIDE AND OUT.

When we designed our new small business computers, we meant business.

As basic as that seems, it is unique. Just about every other microcomputer being sold as a small business system today was originally designed as a kit for hobbyists.

Every design decision was made with *quality* and reliability in mind. The result is dependable performance and a solid appearance for business, professional and scientific applications.

FIRST SMALL SYSTEM WITH BIG SYSTEM STORAGE

Many applications handle large quantities of information, so the DB8/2 uses two quad density 5-inch disk drives with our exclusive Dual Density Disk Controller for up to 1.2 megabytes of formatted storage. That's more capacity than two single density 8-inch drives.

If you need more storage, our DB8/4 has two 8-inch drives with up to 2 megabytes capacity, more than any other dual floppy disk system on the market.

OUR SOFTWARE IS BIG ON BUSINESS

Dynabyte helps you get down to business immediately. The DB8/2 is the first microcomputer to offer enough storage capacity on 5-inch drives to fully utilize CP/M,* the most widely accepted disk operating system. We also supply and support BASIC, FOR-

* CP/M is a trademark of Digital Research.

TRAN and COBOL programming languages. Our applications packages include general ledger, accounts receivable, word processing and many other CP/M compatible programs.

Reliability is a big consideration in buying a business computer, so we built it in. Our edge connectors meet military specifications, the toughest electronics manufacturing standard. Our regulated power supply is designed to meet U.L. standards, which means the entire system runs cool and dependable. And our cast aluminum enclosures are rugged as well as attractive.

AND THE BIGGEST THING OF ALL

Customer support. Our support starts at the factory with testing and burn-in programs that assure the *entire* integrated system is reliable prior to shipment. Our completely modular design allows continuing support in the field. We maintain a bonded inventory of all sub-system modules which means we can deliver replacement sub-assemblies overnight nearly anywhere in the continental U.S.

Dynabyte built in little things, too. Like a fully-populated 12-slot backplane, switched AC outlets for accessories, an option for European power, quiet whisper fans with long-life metal construction, lighted indicator switches for Power On and Halt, a shielded enclosure to protect disk drives from electro-mechanical interference, and a fully enclosed power supply for

operator safety.

Since we didn't cut corners in design, the price/performance ratios of our systems make good business sense.

THE INSIDE FACTS

The DB8/2 Computer System includes two 5-inch disk drives either single or double sided for up to 1.2 megabytes of mass storage; a 4MHz Z-80 processing module with one parallel and two serial ports, an EPROM programmer and up to 4k ROM; 32k of RAM, a 12-slot fully-populated backplane; our exclusive Dual Density Disk Controller, and CP/M.

The DB8/1 Computer includes a 4MHz Z-80 processor with one parallel and two serial I/O ports, an EPROM programmer and up to 4k of ROM; 32k RAM, and a 12-slot fully-populated backplane.

The DB8/4 Disk System, designed to be the mass storage companion to the DB8/1, includes two 8-inch floppy disk drives in either single or double sided configuration for up to 2 megabytes of mass storage, our Dual Density Disk Controller, and CP/M.

All three units will be available in rack mount models.

For a descriptive brochure and price list, call or write Dynabyte, 1005 Elwell Court, Palo Alto, CA 94303. Phone (415) 965-1010.

Or better yet, see your local dealer.

DYNABYTE

YOU CAN DEPEND ON IT.

Z-80 program. George has helped us a great deal in removing some of the bugs that crept into the development of the standard. Mykro manufactures a K.C I/O Cassette Recorder Interface. The interface comes completely assembled with a bonus of the object tape of IAPS. The Mykro interface and IAPS make it possible to get the most out of your system in making totally portable tapes and reading IAPS formatted Floppy ROMs.

The Mykro interface is available for $129.00 by contacting George Sutton, Mykro Corporation, P.O. Box 433, Los Altos, CA 94022 or calling (408) 733-8221.

ANOTHER CONSULTANT ANSWERS CALL

In my August column I requested anyone who is in the consultant business to drop me a note, and I have now been getting responses.

The most recent that we have found is Marvin Mallon of Compu-Quote. Marvin's speciality is business and industrial applications. Working with a number of area dealers, he is able to provide a total service from establishing the hardware to preparing the application. For anyone interested in Mr. Mallon's service, he can be contacted at:

> Compu-Quote
> 6914 Berquist Avenue
> Canoga Park, CA 91307
> (213) 348-3662

I would like to hear from consultants in the midwest and east, so if you're in the business, drop me a note.

GREAT EXPECTATIONS: CAN WE HAVE THEM?

Those of you who were at Personal Computing in Philadelphia this year probably recognize the head as the title of my talk.

Essentially, I was alluding to the impractability of thinking that business systems which will handle all the necessary accounting problems can be obtained for 600 dollars. Unfortunately, the industry has been giving the impression that the general business computer user can expect to have the same level of expectation from a 600 dollar micro as from a 25 thousand dollar mini or a multi million dollar maxi. **It just ain't so!**

Microcomputers are in the third generation of *iron* development but still in the last half of the first generation software. However, there is hope due to the efforts of many of the systems software designers and the growth of application designers. But regardless of how well these people do their jobs, the 600 dollar business machine will not exist.

Consequently, it is extremely important that not only the user educate himself but the vendor of business systems do so also. Pretending that high level capability, flexibility and reliability can be found in less than adequate machines is foolhardy on everybody's part.

The consumer computer is another area in which the industry has been fooling the public. The consumer computer is not a fact or even a probable reality in the foreseeable future. The reason for this is very clear in my mind. Why would anyone buy three thousand dollars worth of hardware to do a job that a ten dollar timer from the hardware store will do equally as well? Or as one reader that I talked to put it: "My wife doesn't even balance the checkbook now, so how can I expect her to do it on a computer?"

Even though I enjoy computers more than most, I really think that putting them into all aspects of daily *life* would be a travesty rather than a help. Already we have become too oriented to relying on so-called work saving devices than on our own initiative. Personally I think it would be better to keep the computer in business and industry and maintain our personal abilities as humans.

—carl

Everybody with brains is buying Radio Shack's $599 personal computer:
the smart new way to save money and time!

Radio Shack's expandable TRS-80™ system is already the best-selling microcomputer in the world. For three smart reasons. People are brainy enough to grasp its usefulness. People find it exceptionally affordable. And because it's available from stock in our stores. It's that simple!

Laboratory work is simplified.

Another customer considers "using the TRS-80 for controlling industrial equipment in the plant." Another has programmed "income tax and plotted Dow Jones averages for last year . . . with no previous computer experience."

Home use? It's like one of the family. Radio Shack's TRS-80 can plan menus, convert measurements, balance bank books, update budgets, tutor the kids in math. TRS-80 will even entertain you — it loves to play games!

TRS-80 reduces office paperwork.

Office use? Businessmen are quick to realize the time TRS-80 saves in paperwork, in updating reports, in bypassing the company's mainframe system. A satisfied CPA writes of creating "a low-cost in-house system that will alleviate our present needs for service bureaus."

Scientific use? TRS-80 is the ideal lab assistant. Program it to catalog specimens, classify drugs, perform the numerous statistical and data manipulation needs of clinical labs.

It's expandable, too — add-on RAM, ROM, plus printers and a disk drive!

Easy to use? Yes! Radio Shack's 232-page programming manual has won plaudits from doctors, lawyers, accountants, technicians, and teachers. If you can type, TRS-80 has a professional full-size 53-key keyboard.

Available in stock at Radio Shack— there's a store or dealer near you!

Service? We've got it. Software? Available, and more to come. Peripherals? Order our printers, disk drive, added memory. Happy customers? By the thousands. Because the benefits — aside from our breakthrough low price — include expandability, field-proven dependability, and consumer confidence in Radio Shack and its parent Tandy Corporation.

ACT NOW! Get details and order today at Radio Shack stores and dealers in the USA, Canada, UK, Australia, Belgium, Germany, Holland, France, Japan. Write Radio Shack, Division of Tandy Corporation, Dept. C-038, 1400 One Tandy Center, Fort Worth, Texas 76102. Ask for Catalog TRS-80. Prices may vary at individual stores and dealers.

Radio /hack®
The biggest name in little computers

Is personal computing worth it?

We want your answers at the NCC '79 Personal Computing Festival. New York City, June 4–7

Has personal computing been worthwhile for you? Every aspect of this fast-growing field is being questioned...from the effort to generalize a subroutine to the cost of the latest hardware. What are your views?

Some key questions about personal computing need answers. How is personal computing enriching our lives and those of our families and associates? What is its potential? What are we getting for our investments in this field? Is it worth the time, effort, cost...even the criticism?

JOIN THE PERSONAL COMPUTING FESTIVAL

You can answer these and other questions by participating in the Personal Computing Festival of the 1979 National Computer Conference, the most comprehensive computer show on earth.

Here's how you can participate:

- Present a paper
- Give a talk
- Organize a panel
- Deliver a tutorial
- Demonstrate your application and equipment

The deadline for receipt of letters of intent to participate is February 1, 1979. Accepted papers will be published in the 1979 *NCC Personal Computing Proceedings.* Honors and prizes will be awarded for the best papers and application demonstrations.

For more details, fill in and return this coupon.

NCC '79

Personal Computing Festival
c/o American Federation of Information
Processing Societies, Inc.
210 Summit Avenue, Montvale, New Jersey 07645
201/391-9810

Send me more details on:

☐ Participating in a Personal Computing Festival session.
☐ Demonstrating my personal computing application.
☐ Keeping me up-to-date on the Personal Computing Festival.
☐ Exhibiting my company's products and services at the Personal Computing Festival.

Name_____

Company_____

Street_____

City _____State_____Zip_____

IA

THE COLUMN

By Sandra Evans, Assistant Editor

It seems impossible to read an article about the future use of computers without becoming caught up in the realm of possibilities rather than actualities. There are so many possibilities in any one field that our lives will certainly be affected greatly by these machines in the coming years. Take a moment to consider the article entitled "The Auto Industry Moves to Microprocessors" by Robert S. Koster and Leslie D. Ball.

Koster and Ball discuss the microprocessor's use in the automobile as a control function and monitoring device. Currently the auto industry is experimenting with ignition and transmission systems. However, the authors make an interesting move from the actualities of engine control into the more thought provoking realm of possibility. Their discussion of alcohol interlock systems, automatic radar brakes and brain wave monitoring devices causes one to lean back a moment and think.

Given the technology, wouldn't it be possible for computers to advance to such a state that all mechanical processes could be monitored and adjusted? Then wouldn't it be a logical step to computerize the auto to deal with all interaction between occupants and car? And finally, if brain wave scanning were a fact, couldn't the car be programmed to react to both human physiology and psychology?

If it could, the car might be programmed to your own particular needs and tastes.

When purchasing a car, you would not only be faced with the normal options of vinyl interior, AM/FM radio, power steering or disk brakes. You would also be confronted with options which could control every movement within the automobile. You could literally create your own personalized car. Naturally the microprocessor would monitor and maintain engine performance. But what about luxury?

Suppose you're the sporty type. Now you order a sports car with dual overhead cams and a racing stripe. But you may one day be able to order a car which could actually simulate racing conditions. Driving the freeway could be like driving the Indianapolis Raceway. Not only would your car be outfitted like a race driver's, it would also simulate speed, vibrate, provide you with squeals in stereo, and turn corners as if you were driving on two wheels.

Behavior modification could also fit in nicely with this idea of personalized driving. If your driving needs to be improved, why not program your auto to zap you with an electrical shock when you make those quick left hand turns or tailgate the car in front of you? And if you maintain the inconsiderate habit of lane changing without signaling, you could program the car to turn on a gentle seat massage when you do remember to use your blinker.

Whatever your image is, whatever your desires are, the car of the future waits for you. The options are yours. How will you design it? □

Can you upgrade to 16-bits on a tight budget?

Of course!

Let me introduce you to the 16-bit world of Alpha Micro. Consider *all* the advantages of upgrading your present 8-bit system with the first fully integrated concept of hardware and software. A concept that provides all the features of a 16-bit minicomputer, yet retains all the flexibility, peripheral options, and low cost of S-100 bus systems.

Now you can have much of the sophistication that was previously restricted to only the mini's...Multi-User, Multi-Tasking, Time-Sharing, Memory Management, Disk-File Management, and yes, Peripheral Independence.

Come on, Join Alpha Micro's 16-bit world...You can do it on an 8-bit budget. Five subsystems let you expand your 8-bit

system capability as your needs, and your budget, allow:

AM-100 16-bit CPU. Two board S-100 bus compatible microprocessor set, complete with software — a year and a half field proven reliability.

AM-200 Floppy Disk Controller. S-100 bus compatible supports PerSci 277 and Wangco 87 disk drives.

AM-300 Six Port Serial I/O. S-100 bus compatible provides six fully programmable RS-232 ports.

AM-400 Hard Disk Subsystem. S-100 bus compatible interface and CALCOMP Trident series drives (25, 50, 80, 200 and 300 megabyte configurations).

AM-500 Hard Disk Subsystem. S-100 bus compatible formatter-controller and

CDC Hawk 10 megabyte hard disk drive.

All Alpha Micro systems are fully software supported including the new completely integrated, minicomputer class accounting system which consists of several hundred programs.

For more information see your local Alpha Micro dealer or write or call.

alpha micro

17881 Sky Park North
Irvine, California 92714
(714) 957-1404

Introducing the NEW STATE-OF-THE-ART LEADER...

16K PLUS

HIGH RELIABILITY MEMORY BOARD FOR THE S-100 BUS

IDEAL FOR CROMEMCO 64K DISC SYSTEMS

Fully meets proposed IEEE Standard

Specially designed for the new high speed disc systems using DMA and Time Sharing

16K PLUS

- **Bank Select**
  This feature allows running up to 512K of RAM and multiple users on your system. It is fully compatible with Cromemco software, using output port 40H.

- **Parallel Addressing Feature**
  You may choose to locate blocks of RAM at more than one address simultaneously. This feature is ideal for mixing North Star software, which begins at 2000H, with other software beginning at 0000H. With parallel addressing you can locate blocks of RAM in the lowest 8K and at some higher address at the same time.

- **High Reliability**
  Reliability begins at Seattle Computer Products with proper design. All inputs to the board have Schmitt triggers which provide superior noise immunity. Next, we select only first quality components for assembly. To catch infant mortality, we test all boards following assembly, fully burn them in, and then retest at full operating speed. Rigorous quality control is used throughout the manufacturing process.

- **Fully Static TMS 4044**
  These Texas Instruments 4K by 1 fully static memory chips require no critical clocks or refresh. They allow a straight-forward, clean design ensuring S-100 bus DMA compatibility.

250 nsec. chips — $445
450 nsec. chips — $410

The 16K PLUS board is offered fully assembled and tested only.

Guaranteed: USA customers — parts and labor guaranteed for one full year. You may return undamaged board within ten days for a full refund (factory orders only — dealer return policy may vary). Foreign and kit purchasers — parts only guaranteed; no return privilege.

Check with your local computer dealer

If our board is not in his stock, he can get one in three to five days.

Factory Orders — You may phone for VISA, MC, COD orders. ($3 handling charge for COD orders only) Purchase orders accepted from recognized institutions. Personal checks OK but must clear prior to shipment. Shipped prepaid with cross-country orders sent by air. Shipping — normally 48-72 hours. Washington residents add 5.4% tax. Spec. sheet, warranty statement sent upon request.

Standard S-100 Memory

Our standard 16K RAM boards using the TMS 4044 have been reduced in price. This is the same board sold worldwide to satisfied customers since January.

	ASM	KIT
250 nsec. chips	$345	$320
450 nsec. chips	$310	$285

Seattle Computer Products, Inc.
1114 Industry Drive, Seattle, WA. 98188
(206) 255-0750

CIRCLE INQUIRY NO. 49

Dear Editor:

After rereading the four parts of Roger Williams' article I still think all I have read is a term paper on computers in business. Lots of "WHY" but very little "HOW".

Part four says all anyone needs is a programmer who is fluent in all the program languages known to man, a systems analyst with a PhD in E.S.P. along with a lifetime each in business and computer systems. System software that can run on anything from a TRS-80 to IBM 370. Applications programs so good they don't need a CPU. Along with all the source code. NUTS.

Why is it on page 16 (August) Dr. Osborne says "no naming names" yet on page 42 TDL and XITAN get it?

Could you have Dr. Osborne put an "X" by those manufacturers, in your advertiser index each month, who are unsound?

Anyone have something to say?

Bob Distler
(805) 487-7422
P.O. Box 6376
Oxnard, CA 93031

Before Bob sent me this letter, he called to discuss his general philosophy regarding the state of the art in general and data processing specifically. One of the questions he brought up on the phone was why everything is disk oriented and did we know of anyone who could give him a "crash" proof argument to support the use of disks and disks only. He also brought up some other ideas and gave us what we feel to be unique insights. Hopefully sometime early next year he will be providing you with some of his expertise in the computer field in the form of a column called RHD, which happens to be his initials and the name of a tape handling program that he has written.

In the meantime Bob would very much like to hear from you either by phone or letter. And believe me it will be a worthwhile experience.

Dear Editor:

A few notes from a satisfied Canadian reader:

1. Items move faster through our mail system if you include the Postal Code. This is that funny series of letters and numbers that appear after the province. My postal code is V6S 1B2. Note that the format is letter, number, letter, *space*, number, letter number. Although this may not be quite as simple as your ZIP code, it does mean that a letter addressed:

   Andrew Bates,
       Canada V6S 1B2

   will be delivered to me. The postal code pinpoints the side of the street in a residential block or even the floor of a building in a business district. How's that for precise!

   Software writers take note: We Canadians need at *least* 6 characters for the postal code and 4 characters for the province (state). And if you are going to check the ZIP for all numbers, please put the check in a subroutine so we can replace it with a suitable check for our postal code.

2. WATTS lines do not cross international borders (at least that is what the telephone operator told me). This means that we people in Canada can't phone you for free like everyone else can. How about letting your people accept collect phone calls from Canada, only so we can use A.G. Bell's famous invention, instead of having to spend hours slugging away at the old typewriter and then waiting for what is an erratic mail service on both sides of the border.

3. Another small request for software writers who are mailing things to Canada: If your package costs $75.00 and is distributed on North Star diskette, for instance, please mark the customs declaration as:

   DISKETTE $6.00
   PRINTED MATTER $69.00

If you mark the price as $75.00 we end up paying duty on the *diskette* as though it cost $75.00. Printed matter comes across the border duty free and there is no duty on an item of less than $10.00 value.

Thanks for listening.

Andrew Bates
Vancouver, B.C., Canada

Andrew, your points are well taken and we would imagine that a number of manufacturers are listening to what you are saying. Yes, the mail is bad and phone rates high, but to get the business a WATTS line would be an excellent idea.

Dear Editor:

Thank you for Floppy ROM™ No. 4 and IAPS™.

After keying program 4 with some corrections my system was able to decode the Floppy ROM. Then the program source file was read as input to BASIC. I am very pleased to say that it worked flawlessly.

My personal use Altair 8800b does not have a disk system yet so I was not able to use the programs. Hopefully future Floppy ROMs may have non-disk applications.

Equipment used:
   Altair 8800b with 24K of memory
   MITS BASIC version 4.1
   MITS 2SIO serial I/O
   Panasonic stereo system
      (receiver with turntable)
   Sony TC 205 portable cassette

A few checksum errors occurred until I advanced the volume control just slightly. No other problems.

John B. Palmer
Boonville, CA

John, we thank you for the kind words, and your letter is representative of a little over 400 we have received to date on Floppy ROM #4 and IAPS. Most of the users appear to like the IAPS idea and have provided us with a great deal of input on how to improve it.

If you were lucky enough to be at PCC '78 in Philadelphia this year, you probably heard Bill Turner, the inventor of IAPS define the complete standard in his talk. If not, we are going to publish the talk and guidebook to IAPS in the November issue. Beginning with the January 1979 issue all Floppy ROMs will be in the IAPS format, plus we will begin asking for software articles to be accompanied by a cassette tape at 300 baud and in the IAPS format.

Dear Editor:

Reference the article by Dr. Jerald L. Ripley, "DEBBI — A User Report" in the June 1978 issue, I too have used DEBBI for several months and would like to question one of Dr. Ripley's problems and point out others.

There are six versions of DEBBI, each designed to run on one of the following microcomputer systems: Intel MDS, SBC 80/10 or 8/20, MITS/Altair, IMSAI, PolyMorphic or Sol. I use the IMSAI (-58) version in a Vector One with iCOM 3712 dual disk drives, but since Dr. Ripley did not state which version he used, some of the following comments may not be valid for his version.

The solution to the first problem mentioned, slow loading from disk due to a listing on the console is to enter Control/O following the DLOAD command. This suppresses all output until another Control/O is entered and is documented on page 5-29 of the DEBBI manual.

DEBBI is definitely a version of MITS/ALTAIR/Micro Soft BASIC, as Pertec now owns both MITS and iCOM. This brings us to what I consider DEBBI's greatest shortcoming, its lack of random files and ability to have open only one input and one output file at any one time. According to one of iCOM's software people, this lack of disk I/O flexibility was left out of DEBBI so it would not compete directly with MITS BASIC in the business applications market.

In summary, DEBBI is an excellent extended BASIC, with lots of whistles and bells, but seems to fall short of a usable disk BASIC. In addition, iCOM's documentation of both DEBBI and FDOS-III has several minor errors and is not as complete as one would wish; however, a letter or phone call to them should get you corrections for all the known mistakes.

I hope that this information is of interest to some of your readers and that Dr. Ripley does not take offense at any of my comments regarding his article.

R.E. Wilson
Dallas, TX

DEBBI is a reasonable extended BASIC, but as we were working on Dr. Ripley's article we found it necessary to call Pertec and ask for their thoughts. They also felt it was a reasonable BASIC but fell short of their expectations as a disk BASIC. It appears that the general feeling among other readers is that there is much better available so why even consider DEBBI at this stage of the game.

Dear Editor:

Can anyone tell me how two or more persons all using their own TRS-80 can play games over the phone lines so each person's video shows the same thing and reacts identically? Please specify how to make or where to buy any additional hardware that may be required.

Also, it's really frustrating to want to use a LI program and can't because you have LII. In most cases, I don't know how to re-do my LI tapes so they'll run on LII. I've got 16K LII and most of my LI tapes still come back with "program too long" after using the conversion tape on them.

Can anyone tell me how to easily change my LII back to LI and vice versa or know of anyone who has developed an inexpensive device to do this without having to bother with LI to LII tape conversions?

Sharon Jackson
P.O. Box 621
Fenton, MO 63026

Sharon, that's a thought provoking idea. Somebody has probably figured out how to set up a communications net for the TRS-80, only we haven't heard about it yet. If any TRS-80 user group has worked on this let Sharon know and us too.

Dear Editor:

As soon as I get your magazine, I read Adam Osborne's *From the Fountainhead* because his candor is much needed in the hobbyist computer market.

But not his June 1978 column. He omitted Step Zero which is crucial and absolutely essential. Step Zero is the question: Do you want a computer for games or for business? If anyone wants a computer for business, then he wants to buy a Wang, a Hewlett Packard, an IBM, etc.

There is no business computer in the hobbyist market.

There is no business computer in the personal market in spite of all of the bally-hoo about the business applications and professional uses.

BEFORE YOU BUY COMPUTER#1, VISIT #1 COMPUTERLAND

If the truth is that you want a computer . . . then we want to be your computer store.

We're ComputerLand, the #1 computer store chain in the U.S. What's meaningful about that fact is, that ComputerLand has been chosen by more people as having what they've been looking for. And, since you're looking, let us tell you what you'll find, when you visit a ComputerLand store.

You'll find a product line that's continually evaluated to provide you with the widest and best selection in quality, brand name microcomputers anywhere. You'll find an enthusiastic and knowledgeable staff able to interpret all the equipment specifications, in terms of how they apply to you, and in a way you'll understand. You'll find demonstration areas where you can get a firsthand experience of running a computer yourself.

COMPUTERS FOR BUSINESS

You'll find educational materials to give you a total insight into the world of microcomputers.

You'll find a fully equipped service department to provide whatever assistance is required to keep your computer running in top-notch condition. You'll find computer user's clubs to join, where you can share ideas with people as enthusiastic as yourself. And, with each new visit, you'll find excitement—from the people you deal with, the equipment they offer, and from your own ever-growing personal involvement.

Enough about us. How about what computers do. To attempt to describe all the things your computer might do, would be to describe your imagination. So instead, we'll briefly list some of the many things for which small computers are already being used.

In business, the advent of the versatile and compact microcomputer has put the benefits of computing within reach of small companies. With systems starting at less than $6000, the businessman can

COMPUTERS FOR THE HOME

computerize things like accounting, inventory control, record keeping, word processing and more. The net result is the reduction of administrative overhead and the improvement of efficiency which allows the business to be managed more effectively.

In the home, a computer can be used for personal budgeting, tracking the stock market, evaluating investment opportunities, controlling heating to conserve energy, running security alarm systems, automating the garden's watering, storing recipes, designing challenging games, tutoring the children . . . and the list goes on.

In industry, the basic applications are in engineering development, process control, and scientific and analytical work. Users of microcomputers in industry have found them to be reliable, cost-effective.tools which provide computing capability to many who would otherwise have to wait for time on a big computer, or work with no computer at all.

COMPUTERS FOR INDUSTRY

And now we come to you, which leads us right back to where we started: **If you want a computer, then we want to be your computer store.**

Whether you want a computer for the home, business or industry, come to ComputerLand first. We'll make it easy for you to own your first computer. Because, simply put, we really want your business. When you come right down to it, **that's what makes us #1.**

ComputerLand®
WE KNOW SMALL COMPUTERS

CIRCLE INQUIRY NO. 9   14400 Catalina St., San Leandro, CA 94577   (415) 895-9363 • Franchise Opportunities Available

The kit computers, the hobby computers, the personal computers, are strictly tinker toys. They are alright for playing Star Trek, blackjack, etc., but no one should attempt to use them for anything serious.

We have 100 manufacturers for the S-100 bus. None talk to the others. It is positively amazing how little of this gross incompatibility of parts is ever mentioned in the computer magazines.

Documentation is either non-existent or stinks.

Service is a joke. Where do you get boards serviced?

My North Star disks went down for five weeks last summer, and again for six weeks this summer. There is no local repair shop. North Star repaired them, but then left the repaired units on their shelf for eight days *after* fixing them, and then shipped them surface UPS, instead of Blue Label Air UPS! Is that any way to run a business?

How do you tell employees that they must wait for six weeks for their paychecks because the payroll program won't work on a DOWN computer?

I put $10,000 and two years of work into my computer — my personal, hobbyist computer. So far, I have had about one good month of use out of it. To base a business on this computer, one would have to desire commercial suicide with passion.

A true commercial computer (Wang, Hewlett Packard, IBM) costs no more than hobbyist junk; it runs as soon as you get it; the documentation is excellent; it can be serviced in a day; and it has business programs ready to run.

Edward L. Tottle
Baltimore, MD

The only thing that you can say when the computer goes down on payday is: "great expectations, can you have them?" Probably not.

Dear Editor:

The Spain Rehabilitation Center at the University of Alabama Medical Center has a project underway to demonstrate both the utility and economic feasibility of the new generation of 'personal' computers for use by the severely disabled. The

programmability of the computer will allow it to serve as a general purpose appliance to be used as an aid to communication and education as well as for environmental control and entertainment.

This system, as currently envisioned, will consist of a microcomputer, an on-line storage device for programs and data, two T.V. monitors for user feedback and information display, a printing device for typed output, a speech recognition device for vocal input of commands, data, and text, a powerline controller for environmental control, and a telephone dialing/answering device. We are attempting to select components which are widely distributed and serviced as well as being plug compatible and economically priced.

Programs will be written or purchased to perform specific functions in each of the four general areas mentioned above. However, we would be very interested in receiving ideas from your readers, particularly those who are disabled, those who have disabled friends or relatives, and those who have personal computers and would like to develop hardware or software for the system on their own, regarding specific functions which they would like to see developed and which could be accommodated by the proposed microcomputer system.

We are sending this letter to several publications and organizations in order to reach as many people as possible and are looking forward to receiving input from anyone who may be interested in this project.

Charles Healey, Research Associate
Spain Rehabilitation Center
U.A.B. University Station
Birmingham, AL 35294
(205) 934-3320

This sounds like a very worthwhile and exciting venture. So since you included us in one of the "many" publications, let's see if some of our readers will supply the necessary input.

Dear Editor:

Help! I am a home brew computer hobbyist who needs an operating system. So I am writing to you in the hope that you or one of INTERFACE

AGE's many readers will please come to my rescue.

I have a home brew '8080' based computer system with a Sykes digital magnetic tape unit that I have interfaced to look like a floppy disk. But I cannot find an operating system to use with it.

Is there any place a home brew computer hobbyist can get a source listing and maybe a paper tape copy of a disk-type operating system to use on a non-standard hobbyist computer system using disk and/or digital magnetic tape?

Glenn Moss
450 N. Mathilda, Apt. Q306
Sunnyvale, CA 94086

Yes, Glenn, there is. Many of the magazines, including us have published complete operating systems that can be customized to meet a specific need. We suggest that you contact Jim Schreier, the man that puts out the SSI Microcomputer Software guide, at SSI, 4327 Grove Street, Phoenix, AZ 85040, and order his book for $7.95. It is the most comprehensive book on available software on the market today. If that doesn't work we have published your full address so other readers can possibly help you out.

Dear Editor:

I read with interest the query of Brother Meyerpeter and your reply concerning the educational uses of microcomputers.

I would highly recommend that you contact Dr. John Hirschbuhl at the University of Akron, Akron, Ohio.

He is by far the leading authority in Computer Aided Education in the United States and most probably, the world.

For your future issue on this subject, John would prove to be your best source for lead articles and the review of other material. His expertise ranges the entire spectrum from psychology of learning and teaching techniques to hardware and software.

John Hodges, President
Kent-Moore Instrument Co.
Pioneer, OH

We suggest all interested parties should also contact him.

Dear Editor:

Re: HELP!

Being a computer enthusiast like many others, I am very anxious in setting up a computer hobbyist/computer user's club with the help of a few of my friends. However, the situation here is not as favorable as might be expected; firstly, most of us lack the necessary technical backing, and secondly, there is a severe shortage of technically qualified personnel who are able or free to help.

Hence I would be most obliged if any of your readers, who are committee members of any clubs/societies, or anybody who might like to help, can provide me with information on how their club/societies were started, how are their meetings carried out/what they do during their meetings, the problems they faced, as well as any other tips and information that might be helpful in the course of setting up a club locally, which may be the first in Singapore.

In anticipation of any form of help anyone might provide, I would like to thank him/her in advance.

Steven Goh
3, Bristol Road,
Singapore 8, Singapore

Steve, let's see if you get any answers.

Dear Editor:

In the last issue we received (April 1978) you mentioned the beginning of the microcomputer in Europe. We would appreciate if you could let readers from your magazine know that we handle some of the U.S. products. We are handling Europe for Meca Alpha-1, also TDL for Holland, Selecterm and Central Data too.

Due to the fact that we buy centrally, we can give our customers the same price as they would have to pay in the U.S., but of course we have to add import duties, etc. This relieves the customers of all these problems.

We are trying to get more software here and we are willing to cooperate with readers in the U.S. to swap information.

J. Boers
Medel B.V., P.O. Box 135
9300 AC Roden, Nederland
Tel: 05908 - 18941

Consider it done.

NOCCC COMPUTER SWAP MEET

The largest computer swap meet *will* be held on Sunday, October 15, 1978 starting at 9:30 a.m. through 3:30 p.m. All computeroids and hobbyists interested in buying or selling should not miss this event.

The Northern Orange County Computer Club (NOCCC) in conjunction with Advanced Computer Products are co-sponsoring this year's swap meet.

It will take place at 1310 E. Edinger in Santa Ana, California. For more information and space reservations call Alice at (714) 558-8813.

EICO DATA PRODUCTS FORMED

A new company specializing in the marketing and distribution of computer terminals has been formed. Called EICO Data Products, it is a division of EICO Electronic Instrument Company, Inc. (O-T-C). Heading the new company is Ms. Linda Ashley whose background includes small business management, education and mathematics.

Ms. Ashley indicated that her company will distribute several types of terminals which will be sold outright or will be available on a lease basis. The terminals can be incorporated into computer systems used by businesses, educational institutions, and personal computers. Terminals serve the function of distributing data-processing information.

For further details contact Ms. Linda Ashley at EICO Data Products, 108 New South Rd., Hicksville, NY 11801 or phone (516) 681-9307.

JEDEC RELEASES CLASS B & C MICROCIRCUIT STANDARD

Responsive to a long-standing need expressed by several user groups, who form the customer base of the semiconductor manufacturing industry, the Joint Electron Device Engineering Council (JEDEC), sponsored by the Electronic Industries Association (EIA), the national Electrical Manufacturers Association (NEMA), and the industry at large, has released JEDEC Publication No. 101 governing JEDEC Requirements for Class B & C Microcircuits.

Publication No. 101, written by the JC13.2 Committee on Government Liaison for Microelectronic Devices, provides an opportunity for OEMs to use a standard quote vehicle and uniform processing spec

which can commonly be used by manufacturer and user alike. The Committee, which has representation from most semiconductor houses across the industry, unanimously approved the spec for use as a standard approach to attempt resolution of the long-standing need from the user community.

The method outlined in Publication No. 101 provides for the use of each manufacturer's own data

sheet, in tandem with standard screening and testing sequences specified in MIL-STD-883 and MIL-M-38510. As an aid to identifying "part" to "spec", a marking standard has also been established which identifies the part by its manufacturer's device type number with a suffix designator JC relating it to the JEDEC sponsored Publication No. 101 specified conditions. Additional marketing specified in

MIL-M-38510 is used to complete the identification for these products.

Copies of the publication are available from EIA's Engineering Department, Standard Sales Office, 2001 Eye St., NW, Washington, D.C. 20006, at a nominal $2.00 per copy.

DUAL IN-LINE LEAD SOCKET PANEL STANDARDS SET

The Electronic Industries Association announces the availability of a new standard, RS-444, "Dimensional and Electrical Characteristics Defining Dual In-Line Lead Socket Panels." A socket panel is a printed circuit board with female contacts inserted through holes in the board. These holes are to receive DIP sockets. This publication establishes a unified numbering system to be used for dual in-line lead socket panels standardized by EIA, and provides standard test methods, gauges and performance requirements for use in the description of these sockets. Performance requirements of sockets described by RS-444 are covered in EIA standard RS-415, "Dimensional and Electrical Characteristics Defining Dual In-Line Type Sockets."

Copies of RS-444 may be ordered at $4.00 each from the Standard Sales Office, Electronic Industries Association, 2001 Eye St., N.W., Washington, D.C. 20006. A free Index of EIA & JEDEC Standards and Engineering Publications is also available upon request.

EIA DEFINES PHONE PLUG AND JACK STANDARDS

The Electronic Industries Association Engineering Department announces the availability of RS-453, "Dimensional Mechanical and Electrical Characteristics Defining Phone Plugs and Jacks." This standard which covers dimensional characteristics and mechanical and electrical values is the culmination of many years' work by the EIA Working Group on Sockets, P-5.2. It is intended to provide standard statements of marking, test conditions, dielectric withstanding voltage, contact resistance, and mechanical dimensions with tolerances in both the inch and metric systems.

Copies of the new standard may be ordered from the Standard Sales Office, EIA, 2001 Eye St., N.W., Washington, D.C. 20006.

STANDARDS ON RACKS, PANELS AND ASSOCIATED EQUIPMENT UPDATED

The Electronic Industries Association Engineering Department has revised RS-310. The most recent revision of this thirty-year-old standard, RS-310-C, "Racks, Panels and Associated Equipment," contains updated dimensions to ensure complete compatibility between racks and electronic gear to be mounted in such racks. This standard should serve as an important communication device between manufacturers and users in the electronics industry. The dimensioning has been based on the positional tolerance (true position) concept and has been given in inches and millimeters to facilitate the conversion to the metric system.

Available at $4.00 each, copies of RS-310-C may be ordered from the Standard Sales Office, Electronic Industries Association, 2001 Eye St., N.W., Washington, D.C. 20006.

TANDY COMPUTER USERS GROUP FORMED

The National Capitol Chapter of the Tandy Computer Users Group has been formed. General membership meetings are held the last Wednesday of each month. The group is open to any and all interested persons. For more details on group activities, you may write to the group President, Rod Wright, 8205 Chivalry Rd., Annandale, VA 22003, or call him at (703) 560-5854.

MINI/MICRO COMMITTEE FORMED

The formation of a Mini-Micro Committee to address the concerns of the developing software products and turnkey system industry using micro and minicomputers was announced by the Software Industry Association of ADAPSO. The purpose of this trade association committee is to work in the interest of member software firms and hardware manufacturers supplying software. Issues planned for discussion include the marketing of software, software support, software protection, technology transfer and training, taxation, pricing, product standards, users groups, plus others that members feel are of general interest and appropriate to the trade association.

Companies interested in further information on the Mini-Micro Committee should contact Stephen M. Hicks, Chairman, Mini-Micro Committee, Forth, Inc., 815 Manhattan Ave., Manhattan Beach, CA 90266, (213) 372-8493.

ATLANTIC RESEARCH OFFERS COURSE ON DATA COMMUNICATION BASICS

Atlantic Research is offering a two-day course on data communications titled, "An Introduction to Basic Concepts and Systems."

The first day of the course will deal with such basics as system components and their functions within the data communication facility; front ends, concentrators, transmission facilities, modems and terminals; the communication channel and its basic capacity; network organization and methods of encoding data onto the communication channel.

During the second day, the course will cover a review of line protocols (Async, Bisync, SDLC); the RS-232/V.24 interface and control of the communication channel, including a review of the control signals and their functions, interaction of the control signals in a typical on-line environment, trouble shooting data communications problems at the RS-232/V.24 interface, and performance monitoring at the RS-232/V.24 interface.

The two-day course costs $250 and is scheduled to be held in a number of cities throughout the United States:

June 19, 20	New York, NY
July 17, 18	Chicago, IL
August 7, 8	Washington, DC
October 16, 17	Washington, DC
December 4, 5	San Francisco, CA

For more information contact Atlantic Research Corp., Teleproducts Div., 5390 Cherokee Ave., Alexandria, VA 22314, (703) 354-3400.

DATA BASE SYSTEMS PUBLISHED

Ronald G. Ross has released a new book, *Data Base Systems.* Published by AMACOM, it is the first comprehensive guide to the still growing field of data base technology.

Ross supplies an introduction to data base management systems as well as a discussion of the evolution of it. He also looks at the direction in which the systems is moving and the practical implementation and management of data base systems.

In addition, *Data Base Systems* describes the various techniques that are currently on the market and looks at the differences between them.

CALENDAR

Nov 1 Columbus Computer Club will meet at the Center of Science and Industry at 7:30 P.M. For further information write c/o Fred Hatfield K8VDU, Computer Data Systems, 1372 Grandview Ave., Columbus, OH 43212, or call (614) 488-3347.

Nov 1 Kitchener Waterloo Microcomputer Club will meet at the University of Waterloo, Room 3388, Engineering Bldg. #4, University Ave., Waterloo, Ontario, Canada at 7:30 P.M.

Nov 1 Lincoln Computer Club will hold its meeting at the South Branch Library located on 27th and South Sts. at 7 P.M. For more details write Hubert Paulson, Jr., 422 Dale Dr., Lincoln, NE 68510.

Nov 1 New England Computer Society will meet in the cafeteria of the MITRE Corp. at 7:00 P.M. Located on Route 62 in Bedford, MA. Contact Dave Day at P.O. Box 198, Bedford, MA 01730, (603) 434-4239 for details.

Nov 1 The Valley Computer Club will meet at 7 P.M. at the Harvard School located at 3700 Coldwater Canyon, Studio City, CA.

Nov 2 Bay Area Microprocessors Users Group (BAMUG) will meet in the Hayward ROC Center, 26316 Hesperian Blvd., Hayward, CA at 7:30 P.M. For further details write BAMUG, 1211 Santa Clara Avenue, Alameda, CA 94501.

Nov 2 Microcomputer Users Group (MCG) will hold its meeting at the University of Minnesota, Electrical Eng. Rm. 115 at 7 P.M. The club meets every Thursday. For more information write MCG, Dept. of Elec. Eng., 123 Church St. S.E., Minneapolis, MN 55455.

Nov 2 Northwest Computer Society meets in the Pacific Science Center in Seattle, Room 200 at 7:30 P.M. The club also meets on the third Thursday of the month. For more details write NCCN, Box 4193, Seattle, WA 98055.

Nov 3 Crescent City Computer Club will hold its meeting at the University of New Orleans, Lakefront Campus at 8 P.M. Call Bob Latham at (504) 722-6321 for more details.

Nov 3 Microcomputer Information Group will meet at 7 P.M. at the Microcomputer Resource Center, 5150 Anton Dr., Rm. 212, Madison, WI 53719, (608) 274-8925. Len Lindsay, president.

Nov 4 Louisville Area Computer Club (LACE) will meet at the University of Louisville, Speed School Auditorium at 1 P.M. For details, write the club at 115 Edgemont Dr., New Alban, IN 47150.

Nov 4 Milwaukee Area Computer Club will meet at 1 P.M. at the Waukesha County Technical Institute, New Berlin, WI. Call (414) 246-6634 for further details.

Nov 4 Oklahoma Computer Club will be meeting at the *Belle Aisle* Library at 10 A.M. Call Al Campbell at (405) 842-4933 for details.

Nov 4 South Central Kansas Amateur Computer Association, 9:00 A.M., Wichita Public Library, Wichita, KS. For further information call Chris Borger at (316) 265-1120 or Dave Rawson, 1825 Gary, Wichita, KS 67219, (316) 744-1629 for further details.

Nov 4 Southern Nevada Personal Computing Society will meet at Clark County Community College, Las Vegas, NV at 12:00. The club also meets on the 3rd Saturday of the month. For further information write SNPCS, 1405 Lucille St., Las Vegas, NV 89101 or call (702) 642-0212.

Nov 5 The Computer Hobbyist Group will meet at 1 P.M. in the Green Center, Rm 2.530, of Univ. of Texas, Dallas. For details write to P.O. Box 11344, Grand Prairie, TX 75051.

Nov 6 Amateur Radio Research and Development Corp. (AMRAD) meets the first Monday of each month at 8 P.M. at the Patrick Henry Branch Library, 101 Maple Ave. E, Vienna, VA. for details write the club at 1524 Springvale Ave., McLean, VA 22101.

Nov 6 Minnesota Computer Society will meet at the Brown Institute, Room 51, 3123 E. Lake Street, Minneapolis, MN. For further information contact the Society at Box 35317, Minneapolis, MN 55435, Attn: Jean Rice.

Nov 7 Tidewater Computer Club will meet at the Electronic Computer Programming Institute, Janaf Office Bldg., Janaf Shopping Center in Norfolk. The club also meets on the 3rd Tuesday of the month. For details contact: C. Dawson Yeomans, Interface Chairman, 677 Lord Dunmore Dr., Virginia Beach, VA 23462.

Nov 8 Home Computers Users Group for Radio Shack TRS-80 meets at 7:30 PM. For details write or call TRS-80 Users Group Information of Eastern Massachusetts, c/o Miller, 61 Lake Shore Road, Natick, MA 01760, (617) 653-6136.

Nov 8 Homebrew Computer Club meeting will begin at 7 P.M. in Menlo Park, CA at the Stanford Linear Accelerator Center Auditorium. Contact the club at P.O. Box 626, Mountain View, CA 94042, (415) 967-6754 for details.

Nov 8 Blackhawk Bit Burners Computer Club meets on the second Wednesday monthly at 7:15 PM in Rockford, IL. For more information contact Frank D. Dougherty, 325 Beacon Dr., Belvidere, IL 61008, (815) 544-5206.

Nov 9 Mid America Computer Hobbyist meeting will be at 7:00 P.M. at Commercial Federal Savings & Loan, Bellevue NE. Intersection of Galvin Rd. and U.S. Hwy. 73-75. Write P.O. Box 13303, Omaha, NE 68113 for further information.

Nov 9 North Florida Computer Society will meet at 227 Edison Dr., Pensacola, FL 32505. For details write this address or call Eugene Rhodes at (904) 453-3844.

Nov 9 The Rochester Area Microcomputer Society will meet at the RIT Campus, Rm. 1030, Bldg. 9 at 7:30 P.M. For details write RAMS, P.O. Box D, Rochester, NY 14609.

Nov 9 Utah Computer Association will meet at Murray High School, Rm 154, 5440 S. State St., Salt Lake City, UT at 7 P.M. For details write or call Larry or Holly Barney, 1928 S. 2600 E., Salt Lake City, UT 84108. (801) 485-3476.

Nov 10 HAUCC will meet at 7:30 PM in Rm 117 of the Science & Research Bldg. of the main campus of the Univ. of Houston. For more details write or call P.O. Box 37201, Houston, TX 77036, (713) 661-6806.

Nov 10 Northern New Jersey Amateur Computer Club (NNJACC) will hold its meeting at the Fairleigh Dickenson University, on the Rutherford Campus, Becton Hall, Room B8, at 7 P.M. For details write NNJACC, 593 New York Ave., Lyndhurst, NJ 07071.

Nov 11 The Permian Basin Computer Group — Odessa Chapter meets at 1 P.M. in the Electronic Technology Bldg., Room 203 on the Odessa College campus. For details contact John Rabenaldt, Box 3912, Odessa, TX 79760, (915) 332-9151.

Nov 12 North Orange County Computer Club will have its meeting at Chapman College, Orange, CA. Doors open at 12:00. 105 Hashinger Hall Auditorium. Membership Chairman, Tracey Lerocker, (714) 998-8080 evenings. For more information write P.O. Box 3603, Orange, CA 92655.

Nov 14 Okaloosa Computer Hobbyist Club will meet in the Community Room of the First Federal Savings & Loan Assoc. of Okaloosa County, 158 Elgin Pkwy N.E., Ft. Walton Beach, FL at 7 P.M. For details call (904) 242-5938.

Nov 14 Rome Area Computer Enthusiasts (RACE) meets on the second Tuesday of every month at Patty's Stagecoach Inn at 7:30 P.M. For details contact Mike Troutman, RD 1, W. Carter Rd., Rome, NY 13440, (315) 336-0986.

Nov 16 Madison Computer Society will meet at 7:30 P.M. at 2707 McDivitt Rd., Madison, WI 53713. Mike Shoh, president.

Nov 16 Sacramento Pet Workshop meets from 7-10 P.M. every third Thursday of the month. For more information contact David Howe, (916) 445-7926.

Nov 17 Amateur Computer Group of New Jersey (ACGNJ) meets at UCTI, 1776 Raritan Rd., Scotch Plains, NJ 07076 at 7 P.M. For further information write to the club at the above address.

Nov 17 Long Island Computer Association meets at 7 PM at the New York Institute of Technology, Old Westbury Campus, Route 25A between Route 107 and Glen Cove Rd., Rm. 508. For more details write Long Island Computer Association, 36 Irene Lane East, Plainview, NY 11803.

Nov 18 Computer Hobbyist Group of North Texas meets at UTA University Hall, Rm 108 at 1 PM in Arlington, TX. For details contact Neil Ferguson at P.O. Box 1344, Grand Prairie, TX 75051, (817) 387-0612.

Nov 18 Philadelphia Area Computer Society will meet at 2 PM at LaSalle College Science Bldg. at the corner of 20th & Olney Ave. For more details write PACS, P.O. Box 1954, Philadelphia, PA 19105.

Nov 18 The 7C's Committee (Affiliated with the Cleveland Digital Group) will meet at Cleveland State University Student Services Bldg., in the Kiva Room at 2:00 P.M. For more information write to Cleveland Digital Group, 8700 Harvard Ave., Cleveland, OH 44105.

Nov 18 San Diego Computer Society will meet at the Grossmont Community College Student Center, 8800 Grossmont College Dr., El Cajon, CA. Doors open at 12:30. For details write P.O. Box 9988, San Diego, CA 92109, or call (714) 565-1738.

Pay a little bit more and get a printer that's brighter than your computer. The BrighterWriter.™

When a few dollars more buys you a first-class impact printer, why settle for a toy? The BrighterWriter gives you quality to start with. And versatility that stays even if you outgrow your present personal computer.

Built smart like the big ones.

The BrighterWriter's a smart printer. There's a microcomputer inside. It outwits even the bigger, higher-priced printers. So you get versatility to do all kinds of printing. And power to grow on.

Prints fat, skinny, tall, small.*

The printer can be as creative as your imagination. Stretch out your characters. Squeeze them close. Make them high. Low. Bold. Banner. You name it.

Plugs into your computer.

No matter what personal computer you own or plan to buy, the BrighterWriter plugs in. Simply and quickly. Hundreds of BrighterWriters are working in Apple, TRS-80, Heathkit, S-100 and many other personal computer systems right now.

Pictures and fancy symbols.*

The BrighterWriter draws out your creativity. You can print drawings, graphs, diagrams, bold symbols, or just about any graphic you can imagine.

Picture your page as thousands of dots. The BrighterWriter can fill in the dots, plot them contiguously, stack them, or scatter them. And its special set of graphic characters simplifies the process.

Prints any character a typewriter can. Faster...

The BrighterWriter can print plain and simple. With 7x7 dot matrix clarity. You get all the letters, numbers, and standard symbols of a regular typewriter. At up to 165 characters/sec.

Ordinary paper.

Fancy or plain, the BrighterWriter prints on ordinary paper. Better yet, it prints on many shapes of paper. Single sheets. Roll. Fanfold.

Want more copies? The BrighterWriter prints multiple copies without extra adjustments.

Four easy buttons.

Operating the BrighterWriter couldn't be simpler. Up-front controls are easy to get to. A power button to turn it on. A test button to self-test your printer. A paper feed button to advance the sheets or forms. A line feed button to advance the paper a line at a time.

Prints any-which-way.

The BrighterWriter comes in two models. The IP-225, at $949, gives you a BrighterWriter with tractor-feed drive for precision forms control. This one can handle everything from labels to 8½" paper widths.

It has eight form lengths and gives you all the features of our IP-125.

A brighter buy.

Our IP-125, friction-feed, BrighterWriter has a 96 character set and prints on 8½" wide paper. Upper and lowercase. It prints expanded characters, too. You can choose a RS-232 serial or parallel interface. $799

Lots of goodies.

There's more. Choose all kinds of options for your BrighterWriter. Up to 132 characters per line, variable character densities, larger buffers, special graphics packages, interface cables, and more.

Give us a call or write. Integral Data Systems, 14 Tech Circle, Natick, MA 01760, (617) 237-7610.

Better yet, see the BrighterWriter at the store nearest you.

Integral Data Systems, Inc.

*Some of these advantages require extra-cost options.

OCTOBER 1978

Nov 19 Central Florida Computer Club will meet at 2010 Fosgate Dr., Winter Park, FL 32789 2:00 PM. Contact Bill Kerns for details.

Nov 19 Cleveland Digital Group meets at 2 P.M. in the old railroad station at Safier's Inc., 8700 Harvard Ave., Cleveland, OH 44105. Write the club at this address for more information.

Nov 21 Rhode Island Computer Hobbyists (RICH) meets the at the Knight Campus of Rhode Island Junior College in the Faculty Cafeteria at 7:30 P.M. For details contact Emilio Iannucillo, RICH, P.O. Box 559, Bristol, RI 02809, or call (401) 253-5450.

Nov 22 Ventura County Computer Society will meet at Camarillo Public Library, 3100 Ponderosa Dr., Port Hueneme, CA 93041 at 7:30 P.M. For more information write: VCCS, P.O. Box 525, Port Hueneme, CA 93041.

Nov 22 Diablo Professional Users Group (DPUG) will meet at Diablo Valley College Library, near the Willow Pass exit of Fwy. 680, from 8-10 PM. For details write or call Bob Hendrickson, Electronics Dept., DVC, Pleasant Hill, CA 94523; (415) 687-8373.

Nov 22 Boston Computer Society will meet at the Commonwealth School, 151 Commonwealth Ave., Boston at 7 P.M. The school is located on the corner of Dartmouth St. in Boston's Back Bay. For information write or call the society at 17 Chestnut St., Boston, MA 02108, (617) 227-1399.

Nov 24 Alamo Computer Enthusiast meets at 7:30 PM in Rm 104 at Chapman Graduate Center at Trinity University, San Antonio, TX. For details call (512) 532-2340, or write to the club at 7517 Jonquill, San Antonio, TX 78233.

Nov 24 Washington Amateur Computer Society will meet at the Catholic University of America, St. Johns Hall, located at Michigan and Harewood Aves. in Washington, D.C. Contact Bill Stewart at (202) 722-0210 for club details between the hours of 10 A.M. and 12 P.M.

Nov 26 Birmingham Microprocessor Group will meet at Southcentral Bell Company headquarters bldg. at 2 P.M. For further details write or call Jim Anderson, 2931 Balmoral Rd., Birmingham, AL 35223; (205) 897-9630.

Nov 26 Summit City Computer Club will meet at the McMillen Library on the Indiana Institute of Technology Campus in Ft. Wayne, IN. For details write the club at P.O. Box 5096, Ft. Wayne, IN 46805.

Nov 28 Southern California Users of RT-11 (SCURT) will meet at 9:30 AM at USC's Annenberg School of Communications. For details call Mark Bartelt, (213) 795-6811, ext. 2663; or Ray Rittenhouse, (213) 640-1830, ext. 225.

Nov 28 Computer Amateurs of So. Jersey will holds its meeting at the National Park Municipal Bldg., 7 So. Grove Ave., National Park, NJ at 7:30 P.M. For details call (609) 541-1010, or (609) 541-8296.

Nov 28 Sacramento Microcomputer Users Group, (SMUG), 7:30-9:30 P.M. at SMUD Training Bldg., on 59 St. Write Richard Lerseth, P.O. Box 161513 or call (916) 381-0335 after 5:00 P.M.

Nov 28 Okaloosa Computer Hobbyist Club will meet in the Santa Rosa Rm, in the Santa Rosa Mall, Mary Esther, FL at 7 P.M. For details call (904) 242-5938.

Nov 28 The Digital Group Group meets the last Tuesday of each month in the meeting room of Consumer Systems at 2107 Swift Rd., Oak Brook, IL at 7:30 PM. For more information write the group c/o William L. Colsher, 4328 Nutmeg Ln., Apt. 111, Lisle, IL 60532.

Nov 28 The Apple Portland Program Library Exchange (APPLE) meets on the last Tuesday of each month at 7:30 PM. For location and details contact Ken Hoggatt, 9195 SW Elrose Ct., Tigard, OR 97223, (503) 639-5505 or (503) 644-0161, Ext. 6136.

Nov 29 The National Capitol Chapter of the Tandy Computer Users Group meets on the last Wednesday of each month. For details contact Rod Wright, 8205 Chivalry Rd., Annandale, VA 22003, (703) 560-5854.

Nov 30 Small Computer Engineering Association of Minnesota (SCEAM) will meet at the Resource Access Center, 3010 Fourth Ave. So., Minneapolis, MN 55408 at 7 P.M. For more information write to this address or call (612) 824-6406.

WHITE COLLAR MICROCOMPUTER

By James S. White

This month's INTERFACE AGE is dedicated to hardware — how you might select your first computer. Various articles discuss topics such as your objectives for using a computer and the equipment necessary to meet these objectives.

Planning for the future is important in selecting and preparing for a computer, at least as important as is planning for other business management functions. Plans are the tracks on which a business runs. A business without plans cannot control its progress any more than can a train off its tracks.

Plans for computer selection and use should support and otherwise be consistent with business plans. Similarly, business plans should include consideration of a computer, what it will require of an organization, and its effects on the organization.

PLAN FOR GROWTH

Planning for the future of computer use is almost always planning for growth. Businesses themselves generally grow, and the growth of a business means growth in the power of the resources, particularly computers, which it needs to operate.

Another factor promoting the growth of computer use is the frequent discovery of new and different additional ways to use the computer. Computers can help in many ways, and success in one application provides a good basis for natural implementation of other computer applications.

Finally, the rapidly increasing capacity and decreasing cost of small computer hardware and software are strong independent factors supporting increases in computer use. The computer is chosen as a tool because it is less expensive than alternatives. As most costs are rising while computer costs are falling, computers are becoming the best solution in more and more applications.

Negative factors influencing growth of computer use are seldom present and even less often significant. An organization may decrease its use of a purchased or leased computer. However, rarely will an organization get a smaller computer just because of unused capacity. The cost of changing to less powerful equipment is generally a greater expense than the resulting savings. Similarly, if one finds that a program has excess capacity, the most cost-effective solution is to leave it unused. Changing programs in such circumstances is rather expensive and rarely results in useful savings.

The impracticality of decreasing computer costs may
be frustrating, but it is realistic. Perhaps this restriction
would become more acceptable if it were considered as
similar to the inflexibility of other semi-fixed equipment
costs. Very often, the rule "what goes up must come
down" just doesn't apply in practical economics, at
least for periods of only a few years.

Recognition of the difficulty of cutting costs may pro-
mote skimping on initial computer investments. This, of
course, is as unwise as splurging. The proper solution is
to plan for needs and opportunities and to plan the most
cost-effective method of meeting these needs and bene-
fiting from these opportunities.

> ...needed increases in capacity
> can be...relatively simple software
> changes...such capacity changes
> need not be...considered in
> planning. However, some capacity
> increases will be very expensive...
> to identify these is important.

THREE GENERAL CONSIDERATIONS

Planning is generally the consideration of change —
how to cause and respond to changes. Three categories
of change may affect computer use. The answer to three
corresponding questions can provide a good basis for
planning the future of a computer in a business.

1. If the environment in which the computer is used
   doesn't change, how will the business' use of com-
   puters increase in the future? Most often, these
   new uses will result from computerizing the more
   obvious present business functions.
2. Change from conditions unrelated to computer use
   is likely in the computer's business environment;
   products, competition, marketing efforts all change.
   How will these changes affect the organization,
   particularly the ways it will want to use computers?
3. How should computers be used to change the
   computer's business environment? For example,
   in a distribution function a computer might be
   selected as an order entry tool, allowing better
   entry of orders and convincing the company to
   aggressively seek small orders which were previ-
   ously unprofitable. Or a computer might allow com-
   puter-supported, exceptionally quick service be
   the basis for its advertised marketing image.

REASONS FOR CAPACITY INCREASES

The need for increases in installed computer capacity
can result from several types of factors specifically
related to how computers are used. Following are the
most common.

First is the need to increase the capacity of the basic
functions which the computer is performing. Examples
might be a business adding products or departments,
thus exceeding the number of products or departments
which the computer hardware and/or software could
handle. In other cases, there may be limits on the number
of orders, customers or employees which can be handled.

Often such needed increases in capacity can be ac-
complished by relatively simple software changes.
Therefore, most such capacity changes need not be spe-
cifically considered in planning. However, some capac-
ity increases will be very expensive to implement, and
planning to identify these is important.

Another reason for capacity increase is the need or desire to use a commercially available software product. Acquiring already programmed software is often much less expensive than writing one's own. However, the software which someone else has written generally is for a computer with features and accessories at least a little different than yours. Therefore, you must add the features the software needs that you don't have.

One may also feel that the computer, or a part of it, is too slow. This reason is often more emotional than rational because there are many ways to compensate for slow computers, ways that often are much less expensive than increasing capacity. Frequently, the slowness is no real problem at all. In any case, planning can help one anticipate the conditions calling for speed increases as well as the time when the need for computer growth may occur.

A more important reason for increasing capacity is the need or desire to do more with one's computer. Having the computer automatically handle more of the manual exceptions required in a computerized system is obviously advantageous, but this type of increase often requires surprising amounts of resources. Allowing more flexibility and more ways to use existing programs is a similar reason for capacity increase. These last two are especially significant to consider when planning because they commonly result from neglecting to include features in early programs which were initially considered unnecessary, but which users later decide they "can't live without".

Consideration of these preceeding reasons can be a worthwhile part of computer planning and yield good results.

WAYS COMPUTERS GROW

What effect on a computer does the need to grow have? One way a computer grows is by adding more memory capacity. More memory will allow larger, more powerful programs as well as the ability to handle more data such as a larger number of customers or employees. Memory capacity can be in RAM or magnetic media such as floppy disks, cassettes, or tape cartridges.

Adding the needed amount of memory can range from easy and inexpensive to impossible. Often a certain amount of memory can be added relatively easily. However, once the normal memory size limit of a particular computer has been reached, further increases are unusually much more expensive. Determining and considering the practical limitations of the computer under consideration can be an important basis for planning.

The other major way of increasing computer hardware capacity is by adding peripheral components. Sometimes the added peripherals can be replacements, perhaps to increase speed. Often they provide totally new functions. Will you want to add voice input or output capability to your computer? Will you want to add on-line controllers, perhaps so your computer can turn on or off a motor? If so, you can benefit from choosing a computer to which such capability can be added with reasonable ease and expense.

PLAN FOR LEARNING

Computer planning is harder than most types of planning, for two reasons. First, the basic technology of computers is changing so rapidly that detailed long range plans are totally unrealistic. The second reason is the great and unimaginable effect that computers can have on your business.

But planning is still possible and vital. A basic premise for the prospective new user's planning is that the first computer will be largely a learning experience. True, one

Heathkit H8 Owners: ARE YOU SERIOUS?

Then upgrade your H8 to a Z80 with the INFO 2000 DISK SYSTEM

Powerful Capabilities

The only full size dual diskette system configured especially for the Heathkit H8 Computer is now available for immediate connection. The INFO 2000 Disk System is the best performer for any microcomputer: it's incredibly fast, it uses full 8" diskettes, and it gives you immediate Z80 capability. The recording format is standard IBM 3740. So you're assured maximum storage capacity, superior error protection, and full interchangeability with other CP/M* based systems.

Simple Connection

The Disk Adapter Board supplied with the INFO 2000 Disk System contains its own Z80 microprocessor. Just swap this board with the 8080 that came with your H8 and you'll have instantly upgraded to a Z80 microcomputer. The Z80 instruction set includes 158—more than twice as many as the 8080. Z80 instructions feature powerful block transfer, block search, block I/O instructions, relative addressing, bit manipulation, index registers and greatly enhanced interrupt processing. All you need to do is plug in the adapter board, cables and AC power, and you're ready to run.

Hassle-Free Operation

Designed just for the H8, the INFO 2000 Z80/Disk Adapter Board includes an EPROM containing PAM-Z. This is a Z80-oriented panel monitor which permits the H8 front panel to operate in both hexadecimal and octal modes. A CP/M disk operating system is included which enables you to use Disk BASIC, FORTRAN, COBOL, PASCAL and hundreds of applications programs. Plus, you can still utilize all Benton Harbor software. The flick of a switch gives you either software mode . . . instantly! You should have a minimum of 16K RAM, and to take full advantage of all the software you can implement, 32K is recommended. You'll also need the standard Heath serial board and a terminal for a full system.

We're Serious

The H8 is a good beginning. But why stop there. Go for the best. With this combination of low cost computer and a powerful disk system, you can now perform a tremendous range of tasks. Use your H8 as a sophisticated business system and a powerful design and development tool.

If you're serious.

Delivered assembled and factory-tested, the complete disk system includes dual diskette drives, Z80/Disk Adapter Board, power supply, cabinet, cables, and CP/M disk operating system. Full price $2950. Available for immediately delivery.

*CP/M is a trademark of Digital Research

INFO 2000 CORPORATION
20630 South Leapwood Avenue
Carson, California 90746
(213) 532-1702

should expect useful results of direct benefit to one's business, and proper planning will ensure such benefits. However, a computer is so different a tool and will have such unanticipated effects on your organization, the best thing to do is to try one and then make fairly specific future plans based on the results of your early experience.

CAVEATS

1. When considering a computer system, *ask the* vendor about the availability and cost of expansion features you may need. Also try to evaluate whether such features may be unavailable when you need them, perhaps because the manufacturer is no longer in business or because he has dropped the feature from his product line.

2. Generally, a doubling of the power of an installed computer is a reasonable plan. Greater increases are often best accomplished by changing to a larger computer. As with any other type of machine, a computer to which too many accessories have been added becomes unwieldy and awkward.

3. As is now obvious, don't buy the top-of-the-line, a computer which can't be expanded, unless you can confidently assume there will be no need to increase capability.

4. Plan ahead, but not too far. About three years is often a reasonable time period: not so far that conditions are unforeseeable, but long enough to allow the first computer to be learned from and proper planning to be done for a second system.

5. Many component manufacturers and equipment vendors will expand their product lines in the future and thus allow the owner of a computer to upgrade his system more than is possible today. However, the buyer who bases his plan on any future additions to product lines is gambling. □

MicroPro International Corporation™

an affiliate of Prodata International Corporation established 1968

professional quality you can count on!

proudly announces

SUPER-SORT©

The ultimate in high performance sort/merge

Specifications

☐ High Performance Tournament/Heapsort Algorithm
☐ Multiple Input Files Read in Parallel for True Merge Operation.
☐ Dynamically Invokable User Exit Routines
☐ Record Selection via SELECT/EXCLUDE Statements
☐ Handles Fixed *and* Variable Length Records
☐ Handles Fixed *and* Variable Length Fields
☐ Handles up to 16 Sort Keys with Intermixed Sequence Indicators and Data Types
☐ Handles Alternate Collating Sequences
☐ Compatible with CP/M* and any Derivative Including ADOS, IMDOS, CDOS, etc.
☐ 8080/8085/Z-80 Compatible
☐ Specialized Optimizations for Floppy Disk Environment
☐ Keyword Command Input For Easy Operator Entry
☐ Benchmarked at Over 560 Records per Minute!
☐ Invokable as a Subroutine from FORTRAN, COBOL and Assembler
☐ Furnished in Relocatable and Executable Form for Easy Load-Address Definition
☐ Optional TAGSORT Operation
☐ Data Types Include ASCII, EBCDIC, Binary, BCD (COBOL packed Decimal), etc.
☐ Supports CP/M-compatible Diskette Files under BASIC, FORTRAN, COBOL and Assembler.

Price $250** includes manual and single density diskette. Manual only $9.00 refundable with purchase.

WORD-MASTER©

The last word in text editing

Specifications

☐ Compatible with any "dumb" CRT possessing addressable cursor and backspace (includes Hazeltine - all models, SOROC, Lear Siegler, IMSAI-VIOC, ADDS Regent, etc.)
☐ Bi-directional word tab, line tab, screen tab
☐ Bi-directional word delete, line delete and character delete
☐ Quad-directional cursor movements
☐ Mid-line insert and delete
☐ Automatic RAM/Diskette buffering with no user intervention
☐ Nested command looping with conditional execution
☐ Global String search, Global String Replace
☐ Queue Buffer for text movement, global replication, and string command storage
☐ Multiple input file merging with user-controlled insertions
☐ Multiple output file control by section under user control

Price $150** includes manual and single density diskette. Manual only $9.00 refundable with purchase.

* CP/M is a trademark of Digital Research

** Prices and Specifications subject to change without notice.

©1978, MicroPro International Corporation. All rights reserved.

Dealer Inquiries Invited: Call (Northern California) (707) 544-2865, (415) 398-7062, (209) 445-0511, (408) 279-8980, (916) 485-7619 (Southern California) (213) 224-1619, (714) 634-2908. Outside California Call Collect (707) 544-2865. Principal offices located at 5810 Commerce Blvd., Rohnert Park, CA 94928

THE MIND REVOLUTION

By Merl Miller

Last month we started a commentary on robot history that ended with the Hopkins Beast. If you haven't read last month's column yet, you may want to now. This month I would like to continue the discussion.

As a result of the atomic energy research carried on in the 1950's, elaborate mechanical arms were developed to handle radioactive materials. They were usually controlled by an operator who was behind a glass shield, some distance from the radioactive material. As the research in this area progressed, the arms became increasingly more sophisticated so that by the early 60's they could be controlled remotely by a closed circuit television system.

A great number of computer scientists began to wonder if these arms could be controlled by a computer. Henry Earnst of MIT was one of the few computer scientists able to obtain one of these mechanical arm devices. He first discovered that the arms were quite agile. They could easily perform fairly complicated functions such as screwing in a light bulb or striking a match, but only if the person operating the machine had his eyes open. If the operator tried to control the arms with closed eyes, little could be accomplished. This meant that if a computer were linked to the machine, it would have to have some "seeing" capability. This was a formidable task. At this time the Hopkins Beast was the most advanced "seeing" robot, and all it could do was "see" an electrical outlet under certain conditions. In addition, pattern recognition was in its infancy.

Dr. Earnst couldn't teach his computer to see so he taught it to "feel". He devised a series of sensors that enabled the arm to feed information back to the computer. Every joint had pressure pads that sensed when the arm touched something, and a series of photoelectric cells helped it distinguish light objects from dark objects. Even with these aids, it could still only fumble a bit. Its most significant accomplishment was a table clearing routine. The arm would systematically sweep back and forth across the table until it bumped into something. When this happened it would try to pick up the object, carry it to the edge of the table and drop it into a bin.

Dr. Earnst's studies showed that although a computer could learn to interact with its environment, it needed a mechanical device specifically designed for computers. It seemed unfeasible to adapt a computer to an already existing machine. Consequently, in the years following the development of Earnst's machine, scientists began putting together complete robot systems that weren't dependent on man being in the loop.

In addition to Dr. Earnst, the most interesting of the modern-day robotic pioneers is Dr. Meredith Thring, head of the Department of Mechanical Engineering at Queen Mary College, London. By clever mechanical design he has developed simple, automatic hands that can pick up a great range of objects including pencils and teacups. He has also designed a group of small vehicles that can safely travel over level or irregular terrain or carry a seated person up and down stairs. One machine can clear a dinner table. Another is quite an amateur "fireman". Its heat-seeking device searches for heat, and when it finds a fire, the machine extinguishes it.

Much of Professor Thring's work is aimed at tele-operator-type devices that are designed as prosthetics. His other devices, such as the mobile fire extinguisher, are special purpose machines with no flexibility in their logic. If you could combine the logical flexibility of a large, general purpose computer with one of Professor Thring's machines, you'd have the first general purpose robot. Now that 16-bit microprocessors are available, this day may not be far off.

I have another thought to share with you. Dave Morris is a computer, electronics and robotics hobbyist in Dallas, Texas. He has a couple years of college and would someday like to get a B.S. in computer science.

I had a very interesting conversation with Dave when I called regarding a letter he wrote to Carl Warren. The letter concerned my July column. What follows is an adaptation of his letter and our conversation, interspersed with some of my comments:

"What if we designed a robot that could drive your car?" That is the question I posed in the July column. The first real problem is how the system differentiates. It will need to recognize a traffic light in the midst of thousands of flashing neon signs, taillights, headlights, turn signals, street lights, store illuminations, light reflections and all the other hundreds of confusing light sources found on a typical city street at night.

The system would also need to estimate human behavior. For example, it would have to decide whether or not a car racing toward your intersection from a side street is going to stop in time. The system will have to look ahead far enough to tell that the weaving car in front of you is, in fact, not avoiding traffic but doing so because the driver is drunk.

Will the system slam on the brakes in the event of emergency or slow for a tumbleweed, paper boxes, dogs or children? And how will we program it to know the difference?

These are all interesting questions and point to the necessity of thinking through all projects.□

"We'll go into mass production and make a fortune."

By Adam Osborne

FROM THE FOUNTAINHEAD

I believe that **serial input/output devices and serial input/output capabilities** will become increasingly more important as telephone communications between microcomputer systems becomes more commonplace. Also, serial I/O devices make very inexpensive controllers for low-speed peripherals such as printers, keyboards, and cassette drives — a frequently overlooked fact. In the last year we have seen a number of new serial I/O devices.

In the past, serial data streams were interpreted using synchronous or asynchronous byte-oriented protocols. The "grandaddy" of all serial I/O devices was probably the Western Digital 1602 which supported asynchronous protocol only. NEC slightly modified the 1602 to generate the μPD369, an asynchronous part, and the μPD379, a synchronous part. But all of these early designs were overtaken by Intel's 8251 and Motorola's MC6850, which offered synchronous and asynchronous protocols on a single chip and were easily designed into a microcomputer system. (Whenever you look at the Intel 8251, always remember that AMD has a much more capable enhancement, the AMD 9551.)

For those of you who are still primarily using asynchronous communications protocol, National Semiconductor's new 8250 ACE is probably the ultimate asynchronous-only part. The 8250 is powerful, easy to interface and easy to program.

For those of you who are primarily using synchronous protocols, the old byte-oriented monosync and bisync protocols have largely been displaced by SDLC and HDLC. These are bit-oriented protocols with a whole new philosophy aimed at increased throughput. Many new synchronous serial I/O parts are available, some of them supporting SDLC only while others support SDLC together with earlier synchronous protocols.

Zilog's Z80-SIO device was the first multi-protocol part to be announced. Unfortunately, the Z80-SIO device still has a few minor "features" which you have to design around; however, it offers two serial I/O channels with asynchronous, synchronous and SDLC protocols all on a single chip. Intel's 8273 is primarily an SDLC part with more SDLC capabilities than the Z80-SIO device but very limited non-SDLC capabilities. The 8273 supports IBM's SDLC loop mode.

A trio of very similar parts are the Fairchild F3846, the Signetics 2652 and the Motorola MC6854. These three parts appear to have been heavily influenced by the same design concepts. All three support SDLC and HDLC protocols, plus (to varying degrees) synchronous byte-oriented protocols. The capabilities of these three parts are, as we might expect, in order of their appearance: the Fairchild 3846 is the most recent and probably the most powerful, the MC6854 comes in the middle, and the Signetics 2652, being the oldest, is possibly the most error-free but the least versatile. The 3846 is the only device with complete and accurate IBM bisync protocol, implemented in chip logic. Before you start using the newest part, remember my well-known warning: "He who buys on the cutting edge of technology shall be sacrificed upon it."

Don't rush into using brand new parts before considering the advantages of using something that is on the verge of becoming obsolete. It will spring no surprises on you.

While on the subject of serial I/O devices and telephone communications, the **Personal Computer Network Committee** (PCNET) are soliciting donations (as a non-profit volunteer organization). Here is a worthy cause for any of you with a little cash to spare. They are looking for donations of $100 for a retaining member, $250 for a sustaining member, $500 for a sponsoring member, and $1,000 for a philanthropic member. I realize most hobbyists do not have enough money to buy the RAM boards they need, but surely there are a few who can help PCNET. If you can, write to:

Personal Computer Network
(PCNET) Committee
701 Welsh Road, Suite 226
Palo Alto, CA 94304

The PCNET Committee's project is possibly the most significant contribution to personal computing that I have seen in the last 12 months. By making available data bases and telephone communications at reasonable cost, they greatly enhance the usefulness of every personal computer. But there is one more reason why I really wish PCNET well: I would like to see them get solidly established with a single protocol before we are faced with ten different incompatible networks that confuse everyone.

I received a letter from Mr. Raymond M. Glueck, singing the praises of **KEA-Microdesign,** their GRAPHIC-ADD board, and the attitude of Mr. Ken Anderson (who is KEA-Microdesign). Mr. Glueck states that he is chief engineer with a large chemical company, which makes his comments all the more meaningful. It is very rare that someone takes the

FROM THE FOUNTAINHEAD
Vectored from previous page

time to write spelling out the good deeds of a vendor; usually they write only to complain. But if KEA-Microdesign is really doing the job that Mr. Glueck suggests, this company should definitely be brought to the attention of all microcomputer users. I would like to solicit comments from KEA-Microdesign customers in particular and comments about the "good guy" companies in general. I have had a flood of critical letters aimed at most manufacturers in the business, and I know that this represents a one-sided sampling. Would you please take the time to write and tell me about the good guys, so that I can publicize their good deeds? Write to me directly at:

Osborne & Associates
P.O. Box 2036
Berkeley, California 94710

or telephone (415) 548-2805.

If you wish to contact Ken Anderson of KEA-Microdesign, his address is:

Box 6531, Station A
Toronto, Ontario
Canada M5W 1X4

I received another interesting letter from Database Computer Systems, P.O. Box 33, Kurait-ONO, Israel.

This company has put together a Technico TMS9900 system for which they are busy generating software. Those of you who may be interested in obtaining **Technico software** should write to Bob Alenco, at Database Computer Systems. □

"Savings Bonds are good for anyone, regardless of income," says Joseph B. Flavin.

THE INVENTOR'S SKETCHPAD

By Roger Garrett
Northeastern Regional Editor

A SYSTEM FOR THREE-DIMENSIONAL GRAPHICS DISPLAY

I have always been fascinated by the idea of three-dimensional television. Unfortunately, most attempts
have involved special glasses worn by the observer, and
the viewer only gets one perspective of the televised image. He looks at a flat screen, and if he walks around the
back of it, he sees the back side of a TV set, not the back
side of the image. Holography (laser photography) may
hold some hope for the future but does not appear feasible at present. I believe, however, that I have an idea for
developing a true three-dimensional system that hobbyists could experiment with.

Figure 1. If the selected point in space (represented by
the LED) corresponds to some part of the 3-D object,
then the LED is turned on, otherwise it is turned off.

Suppose we take a simple LED (light emitting diode) with a power supply and an on-off switch. We mount the LED at some arbitrary point in space. We can then represent a point in space by turning the LED on or the absence of that point in space by turning it off.

Figure 2. The LED on the track is moved quickly back and forth. The computer turns the LED on and off at appropriate points along the track. It "sweeps out" and displays a line in space. The LED is on where the line intersects a surface of the 3-D object and off otherwise. If you move the box the points on the line move correspondingly.

Next, we mount the LED on a movable track so that it can be moved quickly back and forth in a linear fashion. Let's say the range of movement is twenty inches. Instead of a manual switch to turn the LED on and off, we control it with a computer. The computer is set up so that it knows the exact position of the LED along its path at any given moment. We can then store information representing any set of points and line segments into the memory of the computer and, with proper programming, have the computer display those points and lines by turning the LED on and off at the appropriate times as it (the LED) moves along the track. By moving the LED back and forth along its track at a rate of sixty times a second, the human eye does not have time to perceive the individual positions so it sees it as a line or a point. The computer could even make the points and lines appear to move.

For the next step in the development of our system, we substitute a linear array of LEDs mounted on a track which moves the array so that it "sweeps out" a plane of space rather than a simple line in space. Again, each LED is under computer control, and the computer knows the position of the array at any given moment. We store into the computer memory information representing a slice of space, say twenty inches by twenty inches. If, for example, it represented a simple plane through an

AN ARRAY OF LEDs ON THE TRACK

COMPUTER

Movement of the LED array is shown without the track for clarity.

LED ARRAY MOVEMENT

The LED array moves back and forth at 60 cycles per second to "sweep out" a plane in space.

Figure 3. The individual LEDs in the array are turned on and off at appropriate positions in its back and forth motion. The plane through a box appears as a square in the resulting display.

One place (slice) through the object can be displayed. This plane intersects the four surfaces of a box.

empty box, then the information might be a square, and the computer could display it by appropriate control of the LEDs in the array. Just as the points and lines could be made to move in one dimension in the previous setup, the plane figures in this setup could be made to move in two dimensions. It would have the general appearance of a common television set.

Now let's extend the idea further. Suppose we take a two-dimensional matrix of LEDs approximately 1024 by 1024, or, since I have already noted the comparison, a TV

screen and mount it on our track. Again we move it back and forth, but this time our two-dimensional screen is sweeping out a volume of space, i.e. three dimensions. The computer holds information representing many slices of space, perhaps over a thousand such planes, and displays the appropriate one on the screen or LED matrix as the surface moves through its discrete positions along the track. Remember that it is moving back and forth at a very high rate. The result is a three-dimensional image in space.

INTERFACE AGE

The magazine that puts microcomputing in your hands.

Technology has brought the computer within your grasp. **Interface Age** puts it in your hands.

The small businessman like yourself is aware of the progress made in microcomputing.

What you may not have is an update of what is happening in microcomputer development, and what discoveries might save you money and increase your computer's usefulness.

Interface Age has the information you need.

In issues you've missed we've presented...

- A way to improve your chances of success in a business venture.
- A complete general ledger and payroll accounting program with documentation.
- A new design for typewriter keyboards that obsoletes the electric typewriter.
- A program that can assist you in writing reports and letters.
- A program to plan your investment in income property.

Articles ranging from the fundamentals of computers to languages and system design, tutorials, activities, and new product releases to help you get started in microcomputing and keep up with the industry.

Get a hold of the information you need. Subscribe to **Interface Age** today. Save $7 over newsstand price. **12 Monthly Issues: $14 U.S.; $16 Canada/Mexico; $24 International. 24 Issues: $24 U.S.; $28 Canada/Mexico. 36 Issues: $36 U.S.; $42 Canada/Mexico.**

Name_____

Address_____

City_____ State_____ Zip_____

☐ Visa Card ☐ Master Charge

Acct. No. _____ Exp. Date_____

Signature_____
Check or money order (U.S. Funds drawn on U.S. Bank)
Make checks payable to: **INTERFACE AGE MAGAZINE**
P.O. Box 1234 Dept. IA10 Cerritos, CA 90701
☐ *Please send information on issues available back to 1975.*

Two-dimensional matrix of LEDs
1024 x 1024 LEDs. Shown without track for clarity.

There are, of course, some problems with this. The screen must be moved back and forth at least sixty times a second so that the viewer does not perceive image flicker. This has several implications. If we assume that there are 1024 discrete positions of the screen, then each slice of the displayed object must be displayed in one one-thousand-twenty-fourth of one sixtieth of a second. That's a lot of information to display in a very short time. Initial experiments might be done with considerably fewer slices of space and fewer points per slice.

Moving a TV screen twenty inches back and forth at sixty cycles per second would require a substantial amount of energy and would probably result in a lot of little pieces of TV flying about the room. Even an LED matrix would have too much mass to withstand the shaking. The solution is simple.

We position the screen with a lens system that colimates the image upward. This means that if we display an image on the screen and place a flat surface anywhere above the lens and perpendicular to the vertical, the image will be focused on that surface. Now, we could move the flat surface up and down at sixty cycles per second while appropriately displaying the slices on the TV screen, but we will still have a vibration problem.

Suppose, however, that we position a spiral-shaped plate above the lens system. The plate would be made of an opaque material so that an image focused on it from below would be visible from above as well. This plate is enclosed in a solid circular block of clear plastic. The block is then rotated about its vertical axis at 60 revolutions per second with the computer knowing its rotational position at any given moment.

A two-dimensional matrix of LEDs is mounted on the track and moved back and forth.

Figure 4. The LED matrix sweeps out a volume of space. The computer, by turning the individual LEDs on and off at appropriate positions as the matrix moves back and forth, can display any three-dimensional object and even *make it "move" in space.*

CLEAR PLASTIC SOLID

TRANSLUSCENT SPIRAL PLATE

Figure 5. The spiral plate is cast inside a clear plastic cylindrical solid block. The plate is transluscent so that an image projected from the bottom is visible from all angles.

The trick here is that instead of the computer displaying the information associated with *flat* slices of space, it displays the information associated with *spiral* slices of space corresponding to the spiral shape and rotational position of the plate within the clear plastic block. During each rotation, the spiral plate sweeps out the full volume of the block, and by appropriate control of the images on the TV screen, we can display full three-dimensional objects. By utilizing rotational movement, we have eliminated the vibration problems and made it possible to view the displayed object from an angle.

Calculating the appropriate images for spiral slices will prove to be an interesting software problem but should not be unreasonably difficult. The major obstacle will be the timing of the displays and the degree of image resolution possible with today's technology.

SPIRAL PLATE SPINNING AT 60 RPM

FOCUSING, COLLIMATING LENS SYSTEM

Figure 6. The image on the TV screen is collimated by a set of lenses to focus on the spinning spiral plate.

TV SCREEN FACING UPWARDS

One minor drawback that should be considered is that this does not actually reproduce the object, only the *image* of the object. As such, you will be able to see through it. It will appear to be a ghost-like image. You will not be able to, for example, shine a light on it and get a reflection. It should, however, be quite useful in displaying shapes as connected lines. Shading of surfaces may prove quite difficult.

Yet think what an architect could do, constructing three-dimensional models of his design and making dynamic changes to it by simple instructions to a computer. Or how about playing chess with the computer and having the chessboard displayed and movements all made in 3D. Maybe even a Star Trek game could be programmed, with the Enterprise and enemy ships actually flying about in space, firing little beams of light at each other. Sounds fascinating. □

SPIRAL PLATE BLOCK

LENS

TV

Figure 7. Under proper computer control just about any three-dimensional image could be displayed, from architectural designs to Star Trek games.

1978 INDEX
to Microcomputer and
Minicomputer Hardware

What is the microcomputer industry made up of? Hardware, of course. The problem is that within a short three years, the industry has grown from just a few manufacturers to several hundred.

During these years, INTERFACE AGE has provided more pages to listing new products than any other magazine serving the industry. However, we felt that more was necessary, and have answered the problem in the form of an index to available hardware in this issue, to be followed by an index to available software in the November issue. December will bring the third index of available microcomputer books and literature. All three indexes are in response to suggestions from both readers and industry spokesmen, and consequently reflect what we feel to be the desired format.

HARDWARE

The term hardware is meaningless as a word by itself since it can mean anything from a picture hanger to an IBM 370. For the purposes of this industry, hardware are the microcomputer systems, either as a mainframe, or a complete system, or disks and tape units, power supplies, I/O cards and memory cards. Hardware is the physical boxes, cards and wires that make up a computer system. Each piece of hardware performs some function within the unit whole and as such is a separate entity by itself.

Hundreds of articles have been written on the functional purpose of each hardware piece, and even how to integrate them into a complete unit. Within the hardware scheme is the term "iron" which is usually defined as that box which contains all the hardware pieces to be a functional computer of some sort. Therefore, iron, although by definition a piece of hardware, becomes separate as a hardware item.

What this is all leading up to is that hardware and iron refer to different components and are sometimes hard to find or differentiate from, depending upon a person's specific needs. The purpose of this index is not to define each hardware piece as to functional purpose, but to provide a list of what is available to the computer user market and an index to OEMs or system houses that are preparing finished end products.

Table of Contents

MICROCOMPUTER SYSTEMS

• **AB ATEW**
  Box 125, S-642 00 FLEN, SWEDEN
  Telex, 641 20 ATEW S
  LYS 16
    16-bit CPU system — using GPC/P 4-bit slice processors

• **Alpha Micro Systems**
  17875N Skypark North, Irvine, CA 92714
  (714) 957-1404   Attn: Bob Hitchcock
  AM-100
    16-bit microcomputer system

• **Andromeda Systems Inc.**
  14701 Arminta Street, Suite J, Panorama City, CA 91402
  (213) 781-6000   Atten: Les Lazar
  Model 11/B
    LSI-11 based dual floppy system
  Model 11/H
    LSI-11 based cartridge disk system
  Model 11/M
    LSI-11 based mini-floppy system

• **APF Electronics Inc.**
  444 Madison Avenue, New York, NY 10022
  (212) 758-7550   Attn: Neil Lipper
  PeCos I
    Complete system with cassettes and derivative of the
    Rand Joss language.

• **BILLING COMPUTER CORPORATION**
  2000 E. Billing Avenue, Provo, UT 84601
  (801) 375-0000   Attn: National Sales Director

  DC-1204
    Business accounting system CPU, hard disk, printer and CRT

• **CMC Marketing Corporation**
  5601 Bintliff, Suite 515, Houston, TX 77036
  (713) 783-8880   Attn: Bill Tatroe
  TEI PT212/80
    Self contained system, CRT, disk storage, ASCII keyboard

• **Computall Corporation**
  2740 S. Harbor Boulevard, Suite "K", Santa Ana, CA 92704
  (714) 754-7854   Attn: Al Whedon
  CS-20
    Data General commercial sytem printer, disk, CRT
  Micro Nova 9070
    Data General business system complete package
  Micro Nova 9045
    Data General business system complete package
  LSI 4
    Computer Automation complete business system

• **Computer Hardware Inc.**
  4111 North Freeway Boulevard, Sacramento, CA 95834
  (916) 929-2020   Attn: Roger Lotz
  The Time Machine
    Digital cassette/computer system for employee identifi-
    cation and time keeping

• **Computerware**
  830 First Street, Encinitas, CA 92064
  (714) 436-3512   Attn: Sales Manager
  C-BS
    Small business system CRT, 80-132 column printer, floppy
    disk
  C-3D
    Software development system, CRT, 80-132 column printer,
    floppy disk, PROM board, PROM programmer

• **Cromemco Inc.**
  280 Bernardo Avenue, Mountain View, CA 94043
  (415) 964-7400   Attn: Dr. Alice Ahlgren
  Cs-3
    System Three computer, 4MHz Z-80A, 32K RAM, RS 232
    and parallel interface

- **CyberGrafix Advertising Design**
  20201 Staff Street, Canoga Park, CA 91306
  (213) 341-0350   Attn: Shela Clarke
  INFO 2000
  Complete business system

- **Data General**
  Route 9, Westboro, MA 01581
  (617) 366-8911 Ext. 4756 or 4752   Attn: Howard Steiner
  CS/20
  Compact desk top business unit
  MCB 1
  Single board computer 16-bit for data acquisition

- **DATA WORLD Inc.**
  7541 Ravensridge Drive, St. Louis, MO 63119
  (314) 961-2229   Attn: Kenneth Taggart
  VP Series
  Desk top business system, CRT, floppy disk, keyboard, primarily aimed for OEM market

- **Data Terminals and Communications**
  590 Division Street, Campbell, CA 95008
  (415) 326-6141   Attn: Bruce Brough
  TaskMaster
  Total integrated small business system, disks, CRT, console and daisywheel printer

- **Dynabyte, Inc.**
  1005 Elwell Court, Palo Alto, CA 94303
  (415) 494-7817   Attn: Rich Mehrlich
  Basic Controller™
  Single board controller with ZIBL — BASIC like language, Z-80, serial and parallel I/O, cassette I/O, keyboard port
  DB8/2
  Disk based computer system with two 5.25 inch disk drives

- **EXIDY, Data Products Division**
  969 W. Maude Avenue, Sunnyvale, CA 94086
  (408) 736-2110   Attn: Paul Terrell
  Sorcerer™
  Basic system self contained with keyboard, uses ROM cassette, requires video unit

- **General Instrument Corporation**
  300 Shames Drive, Westbury, NY 11590
  (516) 333-9500   Attn: George Weiss
  Series 7000
  Total turnkey system with disk and tape storage system

- **IMSAI Manufacturing Corporation**
  14860 Wicks Boulevard, San Leandro, CA 94577
  (415) 483-2093   Attn: Walt Slater
  VDP-80
  8085 based system 32/64K RAM, CRT, disks and keyboard
  VDP-40
  8085 based system 32/64K RAM, CRT, 5.25 inch disks, keyboard

- **Interact Electronics Inc.**
  P.O. Box 8140, Ann Arbor, MI 48107
  (313) 973-0120   Attn: Michael Tucker
  Model One Home computer
  Built in keyboard, cassette tape deck, requires video unit

- **The Interpring Group Inc.**
  50 Hunt Street, Watertown, MA 02172
  (617) 926-1510   Attn: Sharon Rogolsky
  Summa/11
  LSI-11 based microsystem complete

- **Lear Siegler Inc., Data Products Division**
  714 N. Brookhurst, Anaheim, CA 92803
  (714) 774-1010   Attn. John Pagliaro
  VDP-1000
  Complete data system with CRT and printer

- **Logical Machine Corporation**
  1294 Hammerwood Avenue, Sunnyvale, CA 94086
  (408) 744-1290   Attn: Steven Sester
  ADAM THE YOUNGER
  Small business system, desk top unit, CPU, keyboard, twin floppy disks, printer

- **Micro V Corporation**
  17777 S.E. Main Street, Irvine, CA 92714
  (714) 957-1517   Attn: Art Shahan
  Microstar 25
    Complete desktop business system based on 8085, floppy
    disk controller, two RS232 serial ports.
  Microstar 55
    Multi-user small business system with a data base management system

- **Microproducts**
  1024 17th Street, Hermosa Beach, CA 90254
  (213) 374-1673   Attn: Sales Manager
  Superkim
    Single board control computer — compatible with KIM-1
    software and most APPLE II hardware interfaces.

- **Morrow Computer & Electronic Design Inc.**
  315 Wilhagan Road, Nashville, TN 37217
  Attn: Sales Manager
  SYNAPSE/10
    8080 based microcomputer for data acquisition

- **NOVAL Inc.**
  7044 Convoy Court, San Diego, CA 92111
  Attn: Sales Manager
  NOVAL 770
    Business system housed in wood desk, 48K RAM, CRT,
    dual floppy disk drives and interface to printer

- **North Star Computers**
  2547 Ninth Street, Berkeley, CA 94702
  (415) 549-0858   Attn: T. Burt
  HORIZON
    Microcomputer system with SOROC terminal and floppy
    disks

- **Ohio Scientific**
  1333 S. Chillicothe Road, Aurora, OH 44202
  (216) 562-3101   Attn: National Sales Manager
  C3-B
    Winchester disk based microcomputer system, with triple

processor board 6502A, 6800, and Z-80. Also 48K RAM

- **Olson Electronics**
  Contact any local Olson store
  MP-232
    Business system, 32K memory, CRT, keyboard with numeric
    pad, dual floppy disk drives, dot matrix printer

- **Pertec Computer Corporation, Microsystems Division**
  21111 Erwin Street, Woodland Hills, CA 90049
  (213) 999-2020   Attn: Neil McElwee
  MITS 300
    Complete hardware and software system, hard disk and
    soft disk

- **PolyMorphic Systems**
  460 Ward Drive, Santa Barbara, CA 93111
  (212) 986-6668   Attn: Otto Janssen
  System 8813
    Complete system with floppy disks, CRT monitor and
    keyboard

- **Process Computer Systems, Inc.**
  750 N. Maple Road, Saline, MI 48176
  (313) 429-4971   Attn: Tim Pellegrino
  3800B
    Z-80 based development system — 32K memory, dual floppies, EPROM programmer, CRT terminal and peripheral
    interfaces

- **Quay Corporation**
  P.O. Box 386, Freehold, NJ 07728
  (201) 681-8700   Attn: John Lacatel
  90F/MPS
    Single board Z-80 based microcomputer with floppy controller

- **R2E of America**
  3406 University Avenue S.E., Minneapolis, MN 55414
  (216) 562-9908   Attn: Ronald Larsen
  Micral Cm
    Multiple microcomputer system for multi terminal operations, 8080 based, 16K of RAM, CRT and minifloppies

- **ROCKWELL INTERNATIONAL, Electronic Devices Division**
  5310 Miraloma Avenue, P.O. Box 3669, Anaheim, CA 92803
  (714) 632-2321   Attn: Leo Scanlon
  AIM 65
    Single module microcomputer with an on-board 20 column
    printer
  SYSTEM 65
    Floppy disk based R6500 development system, 2 floppies,
    16K memory

- **RCA/Electro-Optics and Devices**
  Route 202, Somerville, NJ 08876
  (201) 685-6423   Attn: Walt Dennen
  VIP
    Home computer based on RCA COSMAC (CDP1802), 16-key
    keypad, cassette interface

- **Space Byte Computer Corporation**
  6464 Sunset Boulevard, Los Angeles, CA 90028
  (213) 468-8085   Attn: Milt Hubatka
  Space Byte
    Computer business system with dual disks, printer, CRT
    and work station

- **S.D.S. Technical Devices LTD**
  P.O. Box 1998, Winnipeg, Canada R3C 3R3
  (204) 944-1448   Attn: George Sagi
  TDS-M68
    Computer in an attache case, teaching and developmental
    system based on 6800.

- **System Computer and Interfaces**
  223 Crescent Street, Waltham, MA 02154
  (617) 899-2359   Attn: Edward Letscher
  SCI 8010
    Single board computer, 1K RAM, 8K sockets for EAPROM

- **Vector Graphic Inc.**
  31364 Via Colinas, Westlake Village, CA 91361
  (213) 991-2302   Attn: Lori Harp
  Vector MZ
    Z-80 based complete system with dual floppies, 12K PROM/
    RAM board, 32K RAM
  Vector 2
    Z-80 based self-contained desk top microcomputer

- **XITAN Inc.**
  P.O. Box 3087, 1101-H State Road, Princeton, NJ 08540
  (609) 921-0321   Attn: Chris Rutkowski
  GENERAL
    Z-80A based system, 32K RAM, one micro floppy — basic-
    ally word processing system

- **ZEDA Computer Systems**
  1662 W. 820 N., Provo, UT 84601
  (801) 377-9948   Attn: Clair Smith
  Video Computer
    Compact video computer for business or OEM applications,
    Z-80A based, mini-floppy, 48K RAM

- **ZILOG**
  10340 Bubb Road, Cupertino, CA 95014
  (408) 446-4666   Attn: Dave West
  MCZ-1/05
    OEM oriented micro system based on Z-80, with two dual
    floppies

PERIPHERALS

- **Anderson Jacobson**
    521 Charcot Avenue, San Jose, CA 95131
  (408) 263-8520   Attn: Bob Miller
  A 242 A/36
    DEC compatible LA 36 teleprinter acoustic coupler

- **AXIOM**
  5932 San Fernando Road, Glendale, CA 91202
  (213) 245-9244   Attn: Simon Harrison
  EX-801P
    Parallel ASCII input electrosensitive printer
  EX-801S
    RS232/20ma serial input electrosensitive printer
  EX-801H
    9600 serial input electrosensitive printer

- **C. ITOH Electronics Inc.**
  5301 Beethoven Street, Los Angeles, CA 90025
  (213) 390-2668   Attn: Ken Hidaka
  820
  80 column dot matrix impact printer mechanism for 9.5-inch multi ply forms

- **CALCOMP, California Computer Products Inc.**
  2411 West La Palma Avenue, Anaheim, CA 92801
  (714) 821-2541   Attn: Carol Felton
  Model 1012
  Drum plotter desk unit that operates up to 10 inches per second, uses four pens and Z-fold paper

- **CENTRONICS Data Computer Corporation**
  Hudson, NH 03051
  (603) 883-0111   Attn: Chuck Clemente
  Model 765
  1200 bps teleprinter for the high asynchornous transmission speed range

- **COMPUTER APPLICATIONS**
  3030 Bridgeway, Sausalito, CA 94965
  (415) 332-9401   Attn: Arthur Jopling
  MDS-800/PRO-LOG SERIES 90
  PROM programmer adapter to allow direct control and communication with a Pro-Log series 90 PROM programmer

- **COMPUTER TEXTile**
  10960 Wilshire Boulevard, Suite 1504, Los Angeles, CA 90024
  (213) 477-2196   Attn: Sales Manager
  SPRINT 5
  Qume SPRINT 5 daisywheel printing terminal

- **Computronics Engineering**
  7235 Hollywood Boulevard, Hollywood, CA 90046
  (213) 876-3326   Attn: J.B. Stanton
  HEXADECIMAL label keyboard
  16-key zero bounce keyboard for data entry

- **Cromemco Inc.**
  280 Bernardo Avenue, Mountain View, CA 94043
  (415) 964-7400   Attn: Alice Ahlgren

3703
Line printer 180 characters/second, 132 columns, impact printer, bidirectional printing with tractor feed
3355
Daisy Wheel Printer, 55 characters/second, 15-inch platen, tractor feed and friction platen

- **CyberGrafix Advertising Design**
  20201 Stagg Street, Canoga Park, CA 91306
  (213) 341-0350   Attn: Sheila Clarke
  MCD
  SELECTRA-TERM printer
  IOU
  SPINTERM printer

- **Daneva Control Pty Ltd**
  70 Bay Road, Sandringham, Victoria 3181 Australia
  (03) 598-5622   Telex DANEVA 34439   Attn: Stuart Wright
  Duoprint
  64 character, 7x5 dot matrix electrosensitive printer

- **Data General**
  Route 9, Westboro, MA 01581
  (617) 366-8911, Ext. 4755   Attn: Richard Goldberg
  6073/6074 DASHER LP2 printers
  180 cps, parallel receive only line printers

- **Data Printer Corporation**
  99 Middlesex Street, Malden, MA 02148
  (617) 321-2400   Attn: Nick Siedun
  Model 1290
  900 LPM high speed data printer

- **Dataproduct Corporation**
  6219 Desota Avenue, Woodland Hills, CA 91305
  (213) 887-8465   Attn: Gerry Coulter
  2200 Series
  300-600 LPM impact printer
  B-300/600
  300-600 LPM impact printer
  M-200
  200 LPM impact printer
  T-80
  80 LPS non-impact printer

- **Deciter — Division of Jameburg Corporation**
  129 Flanders Road, Westboro, MA 01581
  (617) 366-8334 Attn: Bill Sanford
  26209
  RS232C paper tape reader — desk top model
- **ESMARK**
  507½ E. McKinley Highway, Mishawaka, IN 46544
  (219) 255-3035 Attn: Steve Toussaint
  VIDIET STICK
  Light pen that is compatible with virtually all mini/micro systems, software compatible to 8080 and Z-80 based systems
- **Gentle Electric**
  130 Oxford Way, Santa Cruz, CA 95060
  Attn: Carl Fravel
  Synthesizer/Computer Interface
  Interface to allow audio pickup from voice or instrument to be digitized by a micro system
- **George Risk Industries Inc.**
  GRI Plaza, Kimball, NE 69145
  (308) 235-4645 Attn: Robert Nickels
  Model 721
  ASCII keyboard assembly
- **GSI Systems Corporation**
  223 Crescent Street, Waltham, MA 02154
  (617) 899-6698 Attn: Gerald Gershon
  ITMS-1
  Microprocessor based floppy disk system to replace paper tape, directly replaceable of most paper tape systems
- **Keltron Corporation**
  225 Crescent Street, Waltham, MA 02154
  (617) 894-0525 Attn: Ted Chadurjian
  DM-300
  Digital printer, compact, medium speed unit, 2.5 LPS
- **Lear Siegler, Inc./Data Products Division**
  714 N. Brookhurst, Anaheim, CA 92803
  1-800-854-3805 Attn: John Pagliaro
  300 Series Ballistic™ printer
  Microprocessor controlled dot matrix printer
- **MarComm Inc.**
  124 Tenth Street, Ramona, CA 92065
  (714) 789-3833 Attn: Sales Director
  Sp-100
  5x7 dot matrix bi-directional impact printer
  PROM 68
  PROM programming and associated PROM function interface for 6800 computer systems
- **Nortek Inc.**
  2432 N.W. Johnson Street, Portland, OR 97210
  (503) 226-3515 Attn: Sales Manager
  Dual width module
  Master system clock for use with Digital Equpment Corporation LSI-11, LSI-11/2 and PDP-11/03 computer families
- **OBJECTIVE DESIGN, Inc.**
  P.O. Box 20325, Tallahassee, FL 32304
  (904) 224-5545 Attn: Sales Director
  TRACER
  An S-100 board allows for single stepping through any program
  TIMEMINDER
  An S-100 board with real time clock and hardware interrupts
- **Oliver Advanced Engineering Inc.**
  676 W. Wilson Avenue, Glendale, CA 91203
  (213) 240-0080 Attn: Julie Griess
  UPP-2700
  Stand alone gang PROM duplicator
  PP-series
  PROM programmers for 2708, TMS 2716, 2516 and 2532
  OP-80A
  Optical paper tape reader
- **Periphicon**
  P.O. Box 324, Beaverton, OR 97005
  (503) 646-9869 Attn: Charles Osborne
  Type 511
  Image digitizer

- **PRINTRONIX**
  17421 Derian Avenue, P.O. Box 19559, Irvine, CA 92713
  (714) 549-8272   Attn: Mel Posin
  P600
    600 LPM matrix impact printer
  P300
    300 LPM line printer/plotter
  P150
    150 LPM line printer/plotter
  P300DC
    300 LPM with microprocessor data communications interface for remote printing/plotting

- **Process Computer Systems Inc.**
  750 N. Maple Road, Saline, MI 48176
  (313) 429-4971   Attn: Tim Pellegrino
  SPDS
    EPROM programmer for 8080 and Z-80 systems supports 1702, 2708, 2716 and 2732 EPROMs

- **Rondure Company**
  2522 Butler Street, Dallas, TX 75235
  Attn: R.Shannon
  Selectric Printer
    Parallel 8-bit read only version of ASCII converted selectric

- **SILONICS**
  525 Oakmead Parkway, P.O. Box 9025, Sunnyvale, CA 94086
  (408) 732-1650   Attn: Doug Vaughn
  QUIETTYPE™
    Non-impact ink-jet printer, silent operation at 180 CPS

- **Summagraphics Corporation**
  35 Brentwood Avenue, Fairfield, CT 06430
  (203) 384-1344   Attn: Morris Samit
  BP-11
    Graphics data tablet for S-100 bus machines

- **Sylvanhills Laboratory Inc.**
  P.O. Box 646, Pittsburg, Kansas 66762
  (316) 231-4440   Attn: Sharon Bell
  UNIT-1
    11x17 x-y plotting device console mounted
  UNIT-2
    17x22 x-y plotting device console mounted
  DFT-1
    11x17 x-y plotter kit
  DFT-2
    17x22 x-y plotter kit

DISKS/TAPES

- **The A-Team Inc.**
  P.O. Box 719, Bloomfield, CO 80020
  Attn: Sales Manager
  FLOPPY DISK STORAGE SYSTEM
    Slotted file drawer type device for storing floppy disks

- **Alpha Micro Systems**
  17875N Sky Park North, Irvine, CA 92714
  (714) 957-1404   Attn: Bob Hitchcock
  AM-200
    Floppy disk control for 16-bit machines

- **Braemar Computer Devices Inc.**
  11950 Twelfth Avenue, South Burnsville, MN 55337
  (612) 890-5135   Attn: Richard Morris
  CM 600
    Mini-Dek program loader and I/O device mini cassette

- **BASF Systems**
  Crosby Drive, Bedford, MA 01730
  Attn: J.W. Ehrlich
  6106
    5.25 inch minidisk drive
  6108
    5.25 inch flexydisk, dual sided

- **CALCOMP California Computer Products Inc.**
  1270 N. Kraemer, Anheim, CA 92806
  (714) 632-5461   Attn: Carol Felton
  1143M
    LSI-11 or S-100 bus floppy disk subsystem, IBM 3740 format

- **Cromemco Inc.**
  280 Bernardo Avenue, Mountain View, CA 94043
  (415) 964-7400   Attn: Alice Ahlgren
  PFD-W
  8-inch dual disk drive, 256K bytes per disk, IBM format

- **Dynabyte**
  1005 Elwell Court, Palo Alto, CA 94303
  (415) 965-1010   Attn: Rick Mehrlich
  FD250, FD200, FD650
  Floppy disk drive systems
  D1000
  Fixed disk drive system
  D3000
  Rigid disk drive

- **Galusha Corporation**
  12062 Valley View Street, Suite 220, Garden Grove, CA 92645
  Attn: Sales Director
  MICROSYSTEM/31
  Double density, double sided dual floppy drive system for
  8080, 8085, A-80 and 6800 or 6802 systems

- **Futureworld**
  2514 University Drive, Durham, NC 27707
  (919) 489-7486   Attn: Giles L. Cloninger
  FW-100
  Tape signal conditioner

- **Heath Company**
  Benton Harbor, MI 49022
  (616) 982-3417   Attn: Virgil Bennett
  H17
  102K bytes of floppy storage for the Heath H-8 micro-
  computer system

- **INFO 2000**
  20630 South Leapwood Avenue, Carson, CA 90746
  (213) 532-1702   Attn: Sales Manager
  DISCOMEM
  S-100 bus controller and floppy disk system with CP/M

- **JPC Products Company**
  P.O. Box 5615, Albuquerque, NM 87185
  Attn: G.J. Williams
  IC-3
  Tape cassette interface

- **MECA**
  7026 O.W.S. Road, Yucca Valley, CA 97604
  (714) 365-7686   Attn: Nancy Millican
  Alpha 1
  Digital mass tape storage system
  Delta 1
  Floppy disk and digital tape storage system

- **MICRODATA Corporation**
  17481 Red Hill Avenue, Irvine, CA 92714
  (714) 540-6730   Attn: Richard Yamaguchi
  Lodestar™
  Tape drive system

- **Micropolis**
  7959 Deering Avenue, Canoga Park, CA 91304
  (213) 703-1121   Attn: Charles Ramsey
  Series 1016
  Single and double sided floppy disk system
  1041/1042/1043/1053
  Macrofloppy disk units

- **PCC/Pertec Division**
  9600 Irondale Avenue, Chatsworth, CA 91311
  (213) 999-2020   Attn: Carol Hays
  FD250/FD200/FD650
  Floppy disk drives
  D1000
  Fixed disk drive
  D3000
  Rigid disk drive

- **Smoke Signal Broadcasting**
  6304 Yucca, Hollywood, CA 90028
  (213) 462-5652   Attn: Ed Martin
  BFD-68
  5.25 inch floppy disk subsystem
  LFD-68
  8 inch floppy disk subsystem

- **Tarbell Electronics**
  950 Doulen Place, Suite B, Carson, CA 90746
  (213) 538-4251   Attn: Don Tarbell
  1011
    Floppy disk interface S-100 bus
  1001
    Cassette interface S-100 bus

TERMINALS

- **Beehive International**
  4910 Amelia Earhart Drive, Box 25668, Salt Lake City, UT 84125
  (801) 355-6000   Attn: Dave Zeiter
  MICRO BEE 1
    8085A microprocessor controlled conversational video display terminal
  MICRO BEE 2
    8085A microprocessor controlled buffered video display terminal

- **Bowmar Instrument Corp., Commercial Products Division**
  8000 Bluffton Road, Fort Wayne, IN 46809
  (219) 493-4472   Attn: Tom Utley
  Custom Keyboards
    Multi-colored overlays, custom designed keyboards

- **CMC Marketing Corporation**
  5601 Blintliff, Suite 515, Houston, TX 77036
  (713) 783-8880
  TEI PT208
    Self-contained computer system with display, disk storage, a full keyboard and an 8-slot motherboard

- **Compu-Text**
  287 Wood Road, Braintree, MA 02184
  (617) 848-1800   Attn: Theodore Magida
  CURSOR EDITING TERMINAL
    Cursor editing terminal with high speed inter-system communications and output peripheral switching device

- **Computronics Engineering**
  7225 Hollywood Boulevard, Los Angeles, CA 90046
  (213) 876-3326   Attn: J.B. Stanton II
  HEXADECIMAL LABEL KEYBOARD
    Microprofile keyboard with gold to gold contact system

- **Cromemco Inc.**
  280 Bernardo Avenue, Mountain View, CA 94043
  (415) 964-7400   Attn: Alice Ahlgren
  3100
    CRT terminal
  3101
    CRT terminal

- **Data Access Systems, Inc.**
  100 Route 46, Mountain Lakes, NJ 07046
  Attn: Sales Manager
  DASI 744
    Modified Texas Instruments 743 with switch selectable parity, EIA RS232 interface and cable

- **Datamedia Corporation**
  7300 N. Crescent Boulevard, Pennsauken, NJ 08110
  (609) 665-2382   Attn: Robert Sullivan
  Elite 3045A
    microprocessor-based fully buffered APL/ASCII video terminal with APL overstrike/ASCII underscore

- **Digital Equipment Corporation**
  Maynard, MA 01754
  (617) 493-3716   Attn: Joseph D. Nangle
  VT110
    7x9 dot matrix characters in 80-column, 24-line format, reverse-imaging, split screen, underlining and line drawing graphic characters

- **E&L Instruments, Inc.**
  61 First Street, Derby, CT 06418
  (203) 735-8774   Attn: Sales Manager
  VTE-1
    Video terminal electronic system with full ASCII keyboard, reprogrammable character generator, cursor, and flicker-free refresh

- **Franklin Systems Corporation**
  733 Lakefield Road, Westlake Village, CA 91361
  (805) 497-7755   Attn: Frank Peters
  TO-2000
    Fully automatic operation terminal, automatic sending, full CRT text editing and direct keyboard entry, compatible with all standard Telex/TWX receiving units

- **George Risk Industries, Inc.**
  GRI Plaza, Kimball, NE 69145
  (308) 235-4645   Attn: Sales Manager
  Model 771
    Keyboard subsystem with full ASCII encoding for 71 keys, including separate numeric and cursor pad, auto repeat on all keys, and standard 25-pin interface connector

- **Intertec Data Systems Corporation**
  19530 Club House Road, Gaithersburg, MD 20760
  (301) 948-2400   Attn: Sales Manager
  SuperTerm
    Microprocessor-based printer/terminal with RS232C interface, 60 cps, IBM selectric configured keyboard

- **Lear Siegler/Data Products Division**
  714 N. Brookhurst, Anaheim, CA 92803
  (800) 854-3805/In CA (714) 774-1010   Attn: John Pagliaro
  ADM-31
    Smart terminal with two full 1920 character pages of display with Protect, Write/Product, Program Mode and Cursor Retention
  ADM-42
    Video display terminal with up to 8 pages of memory, 16 function keys, and flexibility of format, editing, interface and transmission
  ADM-3A
    Dumb™ Terminal CRT

- **Megatek Corporation**
  3931 Sorrento Boulevard, San Diego, CA 92121
  (714) 455-5590   Attn: Peter Shaw
  MEGRAPHIC 5014
    Refresh Graphics Terminal compatible with Tektronix 4014, capable of displaying movement, local translation, scale, zoom, selective erase, rotation

- **MicroAge**
  1425 W. 12th Place, Suite 101, Tempe, AZ 85281
  (602) 967-1421   Attn: W. Craig Tenney
  MKB-2
    Keyboard with numeric key pad, upper and lower case, cursor control keys, 2-key rollover, and auto repeat on all keys

- **Motorola Semiconductor Products**
  5005 E. McDowell Road, Phoenix, AZ 85008
  (602) 244-6900   Attn: Sales Manager
  EPIC 68
    Multifunction, display-oriented microcomputer/terminal, 6800 based, serial asynchronous communications

- **MSI Data Corporation**
  340 Fischer Avenue, Costa Mesa, CA 92626
  (213) 393-0622   Attn: Richard Roper
  MSI/88
    Handheld data entry terminal with segmented memory, 16-digit LED display and bar code wand scanning capability

- **The NewO Company**
  246 Walter Hays Drive, Palo Alto, CA 94303
  Attn: Sid Owen
  WRITEHANDER™
    One-handed keyboard for computers, terminals, displays and other 128 character ASCII or ISO coded devices

- **Princeton Electronic Products, Inc.**
  Department H, P.O. Box 101, North Brunswick, NJ 08920
  (201) 297-4448   Attn: Sales Manager
  SYSTEM-850™
    Microprocessor-based computer graphic terminal, or free-standing computer graphic system

- **Ramtek Corporation**
  585 North Mary Avenue, Sunnyvale, CA 94086
  (408) 735-8400
  RM-3000 Series
    Independent display system designed for stand-alone, off-line processing in graphics or display applications

Mini/Micro

COMPUTER CONFERENCE AND EXPOSITION

November 7-8-9, 1978
Astrohall · Houston, Texas

A Major Computer Conference in a Major Computer Market

THE CONFERENCE PROGRAM:

Minicomputers and microcomputers—low cost and versatile—are putting convenient and effective computer power at our fingertips in a vast array of products that will affect every facet of our lives, making minis and micros the fastest growing segment of today's and tomorrow's data processing industry. Designed into systems ranging from traffic lights and numerical control, to paint mixers and kitchen appliances, they offer a new versatility and striking competitive advantages in the end products. We'll examine these aspects—and much more—in the conference rooms at the MINI/MICRO COMPUTER CONFERENCE & EXPOSITION this fall.

Approximately twenty sessions consisting of eighty papers covering both application and design topics are planned.

Some session titles to date would include:

1. The Semiconductor Impact on Computer Systems.
2. Practical Aspects of Audit Control and Security of Minicomputer Systems.
3. Business, Entrepreneurial, and Investment Opportunities in Minis and Micros.
4. From Service Bureaus to In-House D.P. . . . and Vice Versa.
5. Small Business Systems — How to Get Started.
6. The Growing Impact of the Microprocessor in Computer Graphics.
7. Network Computers and Process Control.

8. The Business Aspects of Minicomputer Distributorships.
9. Future Technology for Small Computers.
10. Network Troubleshooting.
11. Minicomputer Tape Subsystems
(12-20). Being Organized

Conference Program Committee — Chairman: Dr. Dennis J. Frailey (Texas Instruments), Drake Lundell (Computerworld), Stanley Klein (Mini-Micro Systems), Neil Kelley (Infosystems), Dr. George Ligler (Texas Instruments).

THE EXPOSITION . . . is growing on a daily basis and is expected to be the largest in the greater Southwest in 1978. The list includes (as of July 10):

Dataproducts, Tektronix, Texas Instruments, CalComp, Centronics, General Automation, Datum, Lear Siegler, Hewlett-Packard, Crea / Comp Systems, Megatek, Honeywell, Ball Electronic, Interdyne, Computer Design, Trilvy, Matrix Publishing, Cipher Data, Small Systems World, Poly Morphic Systems, Tally Corp., Datamation, Summagraphics, Aero Mayflower, Matrix Data Service, EECO, MDB Systems, Control Logic, Computerworld, Printronix, Cap-CPP, Data Systems Design, Cahners Publishing, Digital Equipment Corp., Monolithic Systems, Able Computer, E M & M Computer Automation, Televideo, Intelligent Systems Corp., IPI, Triple I, Computer Divisions, Atlas Energy Systems, Tano Corp., Dataflux

Exhibits Committee Co-Chairpersons:
    Paul Eisner (General Automation
    Linda LaCross (Texas Instruments)

If you design mini-micro computers, sub-systems, peripherals, or components, use them in your business—or plan to—the hundreds of product displays will also be of valuable interest to you.

To: **MINI/MICRO COMPUTER CONFERENCE AND EXPOSITION**
5528 E. La Palma Avenue, Suite 1, Anaheim, CA 92807 Phone: (714) 528-2400

My Primary Interest Is:

Please type or print

☐ **Attending.** Please send me a Preview Program (available early October) listing information on sessions/papers, exhibitors, and hotel reservations.

☐ **Exhibiting.** Please send a copy of the Exhibit Prospectus.

Name_____

Title_____

Company_____

Address_____ M/S_____

City_____ State_____ Zip_____

A Look At Vector Graphic and the MZ

By Carl Warren, Senior Editor

Bob Harp, Chief Engineer

As the microcomputer industry has matured and the market place changed from the hobbyist to the small business user, so has the system design concept of many manufacturers. One of these manufacturers, Vector Graphic Inc. of Westlake Village, California, has recognized the needs of the small business user and has developed a complete system based on the Z-80 CPU. This system is the logical outgrowth of the systems components that have been the mainstay of the Vector Graphic product line.

Beginning two years ago as a supplier of memory boards for S-100 bus type computer systems, Vector Graphic has developed I/O boards, disk systems and now a complete system that utilizes each functional board of the Vector line.

The Vector MZ did not just happen overnight, but was the brainchild of Bob Harp, Vice President and chairman of the board of Vector Graphic. Bob is one of the truly gifted engineers in the microcomputer business today. He received a bachelor's degree in Physics from MIT, his masters and Ph.D. in Electrical Engineering from Stanford. After serving on the faculty of Caltech in Electrical Engineering, he moved to Hughes research where he won the Hyland Patent award for his design work. When asked how long he had been involved in electronics he said: "Ever since I was 10 years old, so about 30 years."

His 30 years of experience have proven to be significant when the quality of design is looked at in the Vector systems. A good example is the mother board used in the MZ. The top surface is a ground plane which reduces the physical amount of distance a signal must travel in the circuit. The result of this, of course, is greater reliability and less noise on the bus. The mother board is fully terminated and is serviced by a 22 amp power supply, which uses a transformer made specifically to Bob's specifications.

A LOOK AT THE MZ

The MZ system sells for $3,750 and comes complete with built-in 5.25-inch floppy disks, Z-80 CPU running at 4MHz, 32K static memory board, shielded and terminated mother board and a professional looking cabinet designed with the small businessman in mind.

The system also comes with Micropolis MDOS and BASIC, plus complete development software including an editor and assembler. CP/M is available at an extra charge. Vector also supplies a MZOS disk operating system which is compatible with software written for the North Star DOS.

Vector is also working on developmental software to assist the OEM and distributor network in developing application software for the end user.

The idea behind the MZ is to provide a complete package in the sense that for the price it is only necessary to purchase a terminal and a hardcopy device. Basically the complete minimum system includes: a CPU, PROM/RAM board, workspace memory, a disk controller, and an I/O board. All of which are found in the MZ system.

Another important area of a complete system is the ease of use. The MZ provides a degree of turnkey operation by providing an immediate reset into the monitor on start up with a simple instruction to boot into the disk operating system. However, the degree of turnkey operation depends upon the applications design and desired interactivity.

APPLICATIONS

The MZ package comes with systems software that can be used by the dealer or end user to develop applications software that meet specific needs. Although Vector employs in-house programmers, the concern is with the systems software and not applications. Vector uses the third party vendor concept for applications software which means that the dealer will be supplying business software either off the shelf or designed specifically for a customer. The MZ lends itself readily to applications

for those users who need a different level of capability in their applications there is the Vector Graphic Mindless Terminal.

This terminal is built around the Ball Brothers video system and uses a capacitance keyboard. The terminal receives its power and intelligence from a video board that is placed directly on the bus. This makes it possible for the CPU to address the screen as memory, and as a result provides protected field capability. Using this type of system, the applications designer can use fill-in-format type application design and better human engineer the software.

Basically all the products that Vector Graphic makes are used in the complete MZ package, but they do have some other unique ideas that will interest another aspect of the microcomputer market.

This other aspect, and one not being addressed by most manufacturers, is the Amateur Radio market. They have developed a prototype digitizer that sits on the MZ bus or any S-100 bus for interpreting slow scan (amateur) television, which makes it possible to save it either on tape or disk media. Although Vector is not directly targeting to the HAM market the idea is to probe all

design since it has flexible operating systems and compatibility to S-100 add ons.

THE MARKET

Vector is primarily addressing the small business market while at the same time providing high quality supplemental boards to the S-100 bus user market.

Vector works through OEMs and a dealer network whom they rely on to provide the level of service to the end user. As Bob Harp put it: "we really can't afford to provide a hand holding operation to the businessman, but we do make every effort to make sure that our dealers have sufficient information and understanding of the MZ system to provide the greatest benefit to the end user."

The marketing and general administrative functions of Vector Graphic is handled by Lore Harp and Carole Ely. Between them they concentrate on ensuring that production and shipping schedules are met and that dealers are kept apprised of new developments.

OTHER DEVELOPMENTS

Along with the MZ system, Vector Graphic offers the Hazeltine terminal to round out the system. However,

facets of micro capabilities and develop useful devices and techniques that further enhance the micro as a business system and ultimately the industry.

WHERE THEY ARE GOING

Vector Graphic is now into their third year of operation and have not found a limit to their possibilities. Bob Harp feels that with 1.75 million small businesses in the country today their growth is assured. He also feels that companies like IBM and DEC are headed toward obsolescence due to technology advances. By this he means that as newer and faster chips are developed, with greater system capabilities, companies like his can quickly provide the finished product. Systems like the MZ, he feels, will not really become obsolete since all that will be necessary to change, in most cases, will be the CPU board. However, to a company like IBM a change to a different technology means a massive change in the total system.

Bob's outlooks are not unlike many others in the microcomputer industry and goes to prove that micros are here to stay. Also that systems such as the MZ will have a major piece of the total market share.□

A SPECIAL FUNCTION APPROXIMATION METHOD AND ITS APPLICATION

By Dr. Endre Simonyi
Office of Consulting Engineers on Judicial Matters
Budapest, Hungary

It very often happens in the engineering practice that one must calculate with functions available in tabulated form. In calculations by computer connected with storing the table in the memory of the computer, the following problems arise:

- If storing happens in the inner memory, the table takes up a great part of the inner memory, and thus little place remains for storing the program and other data.
- Because of the relatively considerable access time of the background memory, when storing in the background memory the use of the table increases the running time of the program significantly.

In the case of both solutions, however, handling the table is an uncomfortable task which can only be realized with difficulty, especially if the number of the variables is higher than two (i.e. the table is more than two dimensional matrix).

Because of these problems, users strive to avoid storing in the tabulated form where possible. That solution could be applied in several cases where the table would be substituted by an approximate function of satisfactory accuracy at which the approximate function could be produced by relatively few program steps and relatively little machine time.

Also, an approximate function must be frequently composed from the description of a mass of facts from some measurement result.

Essentially, both works can be attributed to determination of approximate functions. For producing these functions, certain programs which attempt the approximation by a definite function type are very much in use. If the approximation is not of sufficient accuracy, they attempt by another program and another function type again. The set of function types is only some functions, and trying them is not automatic either. In many cases we do not succeed in achieving a satisfactory accuracy.

Solving this task is especially difficult when the capacity of the computer available is low, or its operating velocity is slight. In this case storing different programs which belong to different functions cannot be solved simultaneously, and the fulfillment of the repeated trials slackens the process to such an extent that it becomes practically unrealizable.

We do not review the programs of certain firms which are related to this article. We refer only to the literature.[1-3] We want to remark only that functions which are used to produce an approximate function are the following:

- linear, parabolic, exponential, hyperbolic, ellipse type.

Harold Balaban worked out a special solution.[4] The solution contains certain elements of our program which is why we deal with it in detail.

The model used is a so-called general linear regression model,

$$g/Y/ = b + mf/X/,$$

and the program also calculates the value of the regression coefficient $/r/$, in addition to the determination of b and m. The program tries the following cases automatically:

$$g/Y/ = f/X/ = \text{linear}$$
$$f/X/ = \text{SQR}/X/,$$
$$f/X/ = \ln/X/,$$
$$f/X/ = 1/X,$$
$$g/Y/ = \ln/Y/; f/X/ = \ln/X/,$$
$$g/Y/ = \ln/Y/,$$

and the maximum number of points is 10. The SQR designation means the squaring. After giving in the data the program prints out the b, m, and r values belonging to the 6 functions automatically without setting up any graduation. This method is an advantage in comparison with the traditional solutions since it is not necessary to feed the data again, function by function. However, the function choice consists of six kinds of functions altogether; the number of variables is two, and the number of the data is considerably limited. Neither the number of the variables nor the function is the user's task.

The application of our program packet has the following advantages:

- Trying over the functions is automatic. Hence, it is not necessary to feed in the data again.
- The number of functions tried over automatically is maximum 1296.
- Selecting the "best" function occurs automatically.
- The number of the data is not so severely limited as in the case of the program reported previously.

Its limit is restricted by the extent of the inner (if it is available) or background memory of the machine.

One special advantage of this program packet is the relative ease in which it is generalized, and this means the number and kind of functions can be changed (increased), the number of variables can be increased, and the data number permitted can be increased by the application of background memory. Naturally it is necessary to take the increase of the operating time into consideration, too.

In the case of two variables, the program uses the following relation for producing the approximate function:

$$G/y/ = b + mf/x/$$

where $f/\ /$, $g/\ /$ mean some kind of function selected from the function set in the program. The program determines the values of b and m to a certain f, g pair by means of the known square-error minimum method. It is a substantial deviation, however, that the formula to be minimized is as follows:

$$\sum_{i=1}^{n} \left[1 - \frac{b + mf/x_i/}{g/y_i/} \right]^2 = \text{min!} \quad /1/$$

At the same time it determines a quantity which serves for characterizing the errors. The determination of the "best" function approximation occurs in such a manner that the program selects an f, g function pair automatically from the function form set available. It also performs the determination of the constants by each function pairs and the value of a quantity serving for the characterization of the errors quoted by S in the following. It selects the function form which can be characterized by the least error by comparing the S values.

The necessary operating time:

$$T = bntm^b \quad /2/$$

where b is the number of the variables (independent and dependent variables); n is the number of data points; m is the kind of number of the function forms in the function set; t is a constant depending on the machine type and the program.

The following relation defines the quantity serving for the characterization of the values of the errors:

$$s = \frac{1}{n} \sqrt{ \sum_{i=1}^{n} \left[1 - \frac{b + mf/x_i/}{g/y_i/} \right]^2 } \quad /3/$$

The s-error function is characteristic of the mean error. The greater its value, the bigger is the mean error, so we look for the minimum of this function.

These programs are four variants:
- two variables /BASIC-4K/,
- two variables /BASIC-8K/,
- four variables,
- two variables and one parameter.

FUNCTION APPROXIMATION WITH TWO VARIABLES /BASIC-4K/

The program:

Line 2-55:	Erase of the screen; print of the heading and legend.
Line 60:	If the memory is 4K byte upon the BASIC interpreter, then

$$N_{max} = 65$$

If your memory is bigger than 4K byte, then you may want to modify this line.

Line 70:	Input number of data points
Line 80:	Print heading of the data columns
Line 90-120:	Input, store and print of the data
Line 130-145:	Initializing of the value of the constants. The function is "good" in this variant if

$$S < \frac{3 \times E\text{-}2}{N}$$

Line 146-151: Compute and store the value of

$$X_{max}; X_{min}; Y_{max}; Y_{min}.$$

Line 152-155: Transform of the value of x,y. The new values are:
$$0 < /x,y/ < = 1$$

The equations are:

$$x_{new} = \frac{x - x_{min} + 1}{x_{max} - x_{min} + 1}$$

$$y_{new} = \frac{y - y_{min} + 1}{y_{max} - y_{min} + 1}$$

Line 160-180:	Compute of the serial number of the function pair.
Line 190-270:	Compute of the value of $F/x_i/$, $G/y_i/$ with the subroutines from the line 500. Store of the value of $F/\ /$, $G/\ /$.
Line 280-320:	Compute of the value of the invalid variable.
Line 340-350:	Compute of the value of A,B-constants.
Line 360:	Compute of the value $S^2 \times N^2$. This line is indexed "S".
Line 370:	Is the new function form "better" than old? If yes, then go to 430.
Line 380:	If no, then: Is not more function form? Yes or no; If yes, then go to 160

$$M_{max} = 100$$

Line 390-415: Compute and print of the results and the name of the "good" function.

Line 430-470: Store of the new constants.

Line 480: Is the function form "good"? Yes or no? If yes, then go to 390.

Line 490: If no, then go to 380.

Line 500-860: Compute of the value of the functions.

Line 515-530: COS/ /-subroutine

Line 541-549: SIN/ /-subroutine.

If

$$P > = \pi/4,$$

then

$$COS/P/ = SIN/\frac{\pi}{2} \cdot P/$$

and

$$SIN/P/ = COS/\frac{\pi}{2} \cdot P/$$

Line 550-560: Linear function.

Line 570-590: Parabolic function.

Line 600-616: LOG/ /-subroutine.

Line 630-650: EXP/ /-subroutine.

Line 660-680: Compute of the square-root with the following equation:

$$SQR/X/ = EXP/\frac{LOG/X/}{2}/$$

Line 690-710: Reciprocal function.

Line 750-820: ASIN/ /-subroutine.

Line 830-860: Compute of the ACOS/ /-function with the following equation:

$$ACOS/X/ = \frac{\pi}{2} - ASIN/X/$$

The mean error of these subroutines:

$$\sim 0.02\%$$

The computing time is:

$$T \sim 10 \times N \qquad /minute/$$

The machine was a SWTPC-6800.

FUNCTION APPROXIMATION WITH TWO VARIABLES /BASIC-8K/

The program is 40% shorter than the 4K-variant. The computing time is:

$$T \sim 3 \times N \qquad /minute/$$

If the memory is 4 Kbyte upon the BASIC-interpreter, then:

$$N_{max} = 80.$$

The program:

Line 2-230: Is not different from the 4K-variant.

Line 240: The name of the function is a string variable.

Line 250-350: Is not different from the 4K-variant.

Line 360: Compute of the value "S".

Line 370-390: Is not different from the 4K-variant.

Line 400-420: Print of the name of the functions is only one line. (25 lines are in the 4K-variant).

Line 430-450: Is not different from the 4K-variant.

Line 460-470: The name of the functions are strings.

Line 480-500: Is not different from the 4K-variant.

Line 520-710: In this version is not subroutine for COS/ /, SIN/ /, LOG/ /, EXP/ /, SQR/ /. This part has 23 lines. (53 lines are in the 4K-version.)

Line 750-860: Is not different from the 4K-variant.

FUNCTION APPROXIMATION WITH FOUR VARIABLES

A detailed review of the program will not be described herein. Figure 1 illustrates a simplified flow chart. In the figure are:

1. data acquisition, store;
2. initializing of the inner constants,

$$S_o/S_o = 1/,S_{min} \text{ and } M/M_o = 1,$$
M-serial number of the constants of the function form;

3. computation of the constants of the linearized system of equations;
4. solution of the linear system of equations;
5. computation of the error function /S/;
6. initialization of the inner constants;
7. printout of the best function form and printout of the values of the constants of the function; and
8. storage of the best function form, S_o, and storage of the values of the constants of the function.

This version is good on a rapid machine only because the machine time is high. For example:

$$T \sim .5 \times N \qquad /hour/$$

was on a Wang 2200 B.

FUNCTION APPROXIMATION WITH TWO VARIABLES AND ONE PARAMETER

The problem is the following: Let us look at some of the serial data points and

$$y = f/x,\alpha/$$

where x,y-variables, α-parameter of serial, and we should like to get

Figure 1.

$$G/y/ = A/\alpha/ + B/\alpha/ \times F/x/$$

approximation function with "good" S-value for all serial.

An example: Function of the electrical resistivity from the temperature:

$$R = f/T, \text{ kind of substance}$$

We search

$$G/R/ = A + B \times F/T/$$

functions, where A,B-functions of the kind of substance. A result is the well known

$$R = R_0 \times /1 + \alpha \times \triangle T/$$

where

$$A = R_0 \times /1 + T_0/$$
$$B = \alpha \times R_0$$
$$G/ \ / = F/ \ / = \text{linear.}$$

My recent method connected with the generalized function approximation program drastically reduces the theoretical work required for setting up the approximating functions usable in practice. The method described by our newest program is as follows:

/a/   Measurement data which is to be described by the relation sought is input.

/b/   The program then chooses the "good" function forms. A function is "good" which produces a greater accuracy than that given beforehand. The computer stores these function forms together with the error values and the constants which belong to the function forms and the measurement data, respectively.

/c/   Data belonging to the next type of substance is input.

/d/   The program then investigates the function forms found to be "good" for the first substance type. It selects those from the ones which give a "good" value for this material. The stored data are identical with the data described in /b/. We then repeat those written in the /c/ and /d/ for all substance types.

/e/   Finally, the value of the constants and the errors belonging to the single measurement data for all the materials are calculated.

Thus, setting up an approximating function simplifies into feeding in the measurement data and reading the result. We do not review the application of the programs; we refer only to the literature.[5-9] □

REFERENCES

[1]520/600 Series Vol. 1. General Library Wang Laboratories, Inc. 1972.
1000-2-ST3, 1001-2-ST3, 1002-2-ST3, 1004-2-ST3, 1005-2-ST3 Programs.

[2]TEK-31 Statistics Program Library Section 4 Curve Fitting 4-1, 4-2, 4-3, 4-4, 4-5 Programs.

[3]Handbook of Quality Control Programs. Litton Business Systems Inc. 1972. 4022MS, 2006MS Programs.

[4]SWAP Program Library 1974. S. 199-7.8.

[5]Racz V., and Simonyi E.: Some Applications of the Programmable Desk-top Calculators in the Clinical Laboratories. IMSZI Kozlemenyek 1974. 11.1.

[6]E. Simonyi: A Special Function Approximation Method and Its Application in Chemical Engineering Processes. IMSZI Scientific Publications 5. 1975.

[7]Uzsoki-Simonyi-Varnay: IMSZI 111/73. szakvelemeny 1973. nov. (In Hungarian)

[8]Varadi-Simonyi-Serege: IMSZI 206/1974. sz. szakvelemeny 1975. maj. (In Hungarian)

[9]E. Simonyi: A Special Function Approximation Method and Its Application. March 1977, Vol. 11, No. 1, 12-15.

PROGRAM LISTING

```
0002 REM
0005 PRINT TAB CHR$(16),CHR$(22)
0010 PRINT TAB(13), FUNCTION APPROXIMATION WITH TWO VARIABLES
0020 PRINT
0030 PRINT FUNCTION FORM: G(Y)=A+B*F(X)
0031 PRINT          WHERE: G( ),F( )-FUNCTIONS
0040 PRINT                 X,Y-VARIABLES
0041 PRINT                 A,B-CONSTANTS
0045 PRINT  RELATIVE STANDARD ERROR: S=SQR(SUM(1-(B*X+A)/Y)↑2)/N
0050 PRINT
0055 PRINT  N-NUMBER OF DATA POINTS
0060 DIM X(80),Y(80)
0070 INPUT N
0080 PRINT X , Y
0090 FOR I=1TON
0100 INPUT X(I),Y(I)
0110 PRINT X(I),Y(I)
0120 NEXT I
0130 S0=1
0140 M=0
0141 S1=1E-3
0142 X1=1
0143 X2=0
0144 Y1=1
0145 Y2=0
0146 FOR I=1TON
0147 IF X(I)<X1THENX1=X(I)
0148 IF X(I)>X2THENX2=X(I)
0149 IF Y(I)<Y1THENY1=Y(I)
0150 IF Y(I)>Y2THENY2=Y(I)
0151 NEXT I
0152 FOR I=1TON
0153 X(I)=(X(I)-X1+1)/(X2-X1+1)
0154 Y(I)=(Y(I)-Y1+1)/(Y2-Y1+1)
0155 NEXT I
0160 M=M+1
0170 J1=M-10*INT((M-1)/10)
0180 J2=1+INT((M-1)/10)
0190 FOR I=1TON
0200 P=X(I)
0210 J=J1
0220 GOSUB 500
0230 U=P
0240 D$=H$
0250 P=Y(I)
0260 J=J2
0270 GOSUB 500
0272 IF ABS(P)<1E-20 THEN P=1E-20
0274 IF ABS(U)>1E10 THEN U=1E10
0280 C1=C1+U/(P*P)
0290 C2=C2+(U*U)/(P*P)
0300 C3=C3+U/P
0310 C4=C4+1/P
0320 C5=C5+1/(P*P)
0330 NEXT I
0335 IF C5*C2=C1*C1 THEN C5=C5*1.0000001
0340 B1=(C5*C3-C1*C4)/(C5*C2-C1*C1)
0350 A1=(C4-B1*C1)/C5
0360 S=SQR(ABS(N-2*B1*C3+B1*B1*C2-2*A1*C4+2*A1*B1*C1+A1*A1*C5))/N
0370 IF S<S0THEN430
0380 IF M<=100THEN160
0390 PRINT  N= ;N, A= ;A, B= ;B, S= ;S0
0400 PRINT F( )= ;F$, G( )= ;G$
0420 STOP
0430 S0=S
0440 A=A1
0450 B=B1
0460 F$=D$
0470 G$=H$
0480 IF S0<S1THEN390
0490 GOTO 380
0500 ON J GOSUB520,542,550,570,600,630,660,690,750,830
0510 RETURN
0520 H$= COS( )
0530 P=COS(P)
0540 RETURN
0542 H$= SIN( )
0544 P=SIN(P)
0546 RETURN
0550 H$= ( )
0560 RETURN
0570 H$= ( )↑2
0580 P=P*P
0590 RETURN
0600 H$= LOG( )
0610 P=LOG(P)
0620 RETURN
0630 H$= EXP( )
0640 P=EXP(P)
0650 RETURN
0660 H$= SQR( )
0670 P=SQR(P)
0680 RETURN
0690 H$= 1/( )
0700 P=1/P
0710 RETURN
0750 H$= ACOS( )
0760 K=9
0770 L=P
0775 O=0
0780 FOR R=2TOK
0790 L=((2*R-3)*(2*R-3)*P*P*L)/((2*R-2)*(2*R-1))
0800 O=O+L
0810 NEXT R
0815 P=O+P
0820 RETURN
0830 H$= ASIN( )
0840 GOSUB760
0850 P=355/226-P
0860 RETURN
```

The Auto Industry Moves to Microprocessors

By Robert S. Koster, M.B.A.
and Leslie D. Ball, Ph.D.

During the last several years the automobile has changed dramatically. The changes have occurred as a result of increases in government regulation which impact on the performance of the vehicle. To a lesser extent, changes in American driving habits have caused some of the changes to occur.

While the automobile has changed, so have computers. In the early 1950's, several automobiles might fit inside of a computer, while today several computers would be easily placed inside of an automobile. Computer technology has moved to miniaturization and has increased the areas in which they might be employed.

It is not surprising, then, that the automotive industry has looked to computers to assist them in meeting government regulations. In this article the authors describe microprocessors which are the technology that allows the automotive industry to incorporate computers in current and future designs. In addition, the authors review what those current uses of microprocessors are and how we might expect the automotive industry to employ them in the future.

WHAT ARE MICROPROCESSORS?

A microprocessor is the central processing unit (CPU) of a computer, reduced in size to fit on a single silicon chip. Its functions, like those of a larger CPU, are to receive data, store it for processing, perform arithmetic and logic operations, and to output results. With the addition of some input/output chips and more memory, a microprocessor is transformed into a microcomputer. The distinction between a processor and a computer is often blurred, and frequently, the terms are used synonymously.

The microprocessor owes its existence to the technology of large scale integration. This technology allows the incorporation of thousands of electronic components in the space formerly occupied by only one or two components. Like many of its predecessors, such as transistors, the microprocessor is being hailed as one in the chain of electronic miracles that has and will continue to have a profound influence on our lives. Because they are so small and inexpensive, the microprocessor is being used in many previously unheard of applications from intelligent instruments to electronic games.

For many applications, the value of a microprocessor is the built-in control function that it can add. It can provide a very sophisticated, yet inexpensive, feedback loop which allows it to be incorporated into many noncomputing devices.

Also, since the microprocessor serves as the guts of a microcomputer, it can bring computing down to a highly decentralized or distributive system. In these systems, each user can have his own computer and data base which communicate with each other rather than just employing terminals tied to a large computer. The use of microcomputers can certainly be compared to the history of electronic calculators since as the cost is reduced, their use and importance will increase significantly.

WHY MICROPROCESSORS AND AUTOMOBILES?

In automobiles, microprocessor use can be most easily traced to the government mandate that motor vehicles meet very strict emissions and fuel economy standards. While the regulation of *either* pollution or economy may not have individually pushed the automotive engineers

to microprocessors, the combination of these two contradictory goals has required a degree of control that is not possible with previously employed mechanical means.

The microprocessor provides the feedback mechanism to insure that the engine is running at the efficiency level necessary to comply with government standards. Because the microprocessor is constantly finetuning the engine, it can maintain optimal performance, even as parts are wearing and the environmental conditions are changing. This heuristic feature is significant because government regulations are leaning to maintaining standards rather than just meeting them at the time of sale.

While it might be unfair to give the federal government all the credit for the use of microprocessors in automobiles, it is quite unlikely that computer technology would be finding its way into cars so quickly without government action. This is especially true since minimizing engine emissions is not cost reducing or inherently marketable. Without strict government regulation, microprocessors would slowly find their way into luxury vehicles for driver convenience and comfort. It is quite probable that as costs were reduced and more applications discovered, that microprocessors would find their way into cars in many different functions. What the government has done is to accelerate development and concentrate that development in areas believed by Congress to be important.

IGNITION SYSTEMS

Although computer technology will find its way into many automotive systems in the future, the most immediat application is the ignition system. The Oldsmobile Toronado is the only production automobile currently equipped with a microprocessor. By 1981, the year of stringent emissions controls, the use of microprocessors will be widespread.

The general function that a microprocessor will perform is to control the timing of the spark. The Delco-Remy Microprocessed Sending and Automatic Regulation System offered in the Oldsmobile does what its name implies. It will conrol spark timing for all load and speed conditions consistent with driveability while complying with all emission control requirements. This system has four sensors that input crankshaft position, manifold vacuum, coolant temperature, and reference timing. As these inputs do not tax the capacity of the microprocessor, other control functions could be added to the system later.[1]

Ford will introduce a microprocessor system in its Versailles V8 5-litre engine to control spark timing and exhaust gas recirculation. Buick has introduced a closed loop knock-limiting system on their 1978 V6 Turbocharged 3.8-litre engine. The Buick system is an analog system which employs one sensor that actually hears engine detonation and sends a signal to retard the spark. This system is quite different from the microprocessor systems because it functions only as a knock limiter and has no other control features.[2]

Chrysler is presently developing a microprocessor to replace its present analog lean burn spark timing system. The system, developed for Chrysler by Texas Instruments, Inc. and RCA Corporation, accepts data from several different inputs to control emissions. These inputs are: ambient air temperature, throttle position, throttle rate of change of position, crankshaft position, intake manifold vacuum, engine coolant temperature, and inlet air temperature.[3] The microprocessor digests the information and adjusts the timing accordingly. Mechanical means do not provide for this level of control, but until quite recently, there had been no motivation to precisely control spark timing.

The microprocessor does not actually perform engine timing; it merely maintains it at optimum levels. This is a large distinction. When the microprocessor acts as a monitor of engine performance, its failure would not prevent the engine from operating. It would only reduce its performance to that achieved from mechanical control. If the microprocessor performed the engine timing, a failure would prevent the engine from working.

This monitoring function also greatly reduces the amount of data that the microprocessor must process. As a monitor, it is only comparing actual performance with the ideal and making corrections. This function requires only a few commands every second, rather than the hundreds required if it actually performed the function.

SENSOR TECHNOLOGY

Electronic engine control systems depend on sensors to measure environmental factors and report this infor-

mation back to the control unit. Since the sensors measure, they are analog in nature. This analog signal must be converted into digital form in order to be processed by the microprocessor. A special interface circuit translates the sensor's analog signal into the microprocessor's digital language. Design of this interface is critical because anticipated changes in the sensor must be incorporated into the microprocessor chip.

While technology in digital circuits is constantly reducing microprocessor costs, similar advances in sensor design must be accomplished to reduce total system costs to levels acceptable to the automotive industry. As all microprocessor based engine control requires sensors to gather information, the limiting factor is the cost and reliability of the necessary sensors and not microprocessor considerations.[4]

Automotive sensors measure five different functions: temperature, pressure, position, fluid flow, and environmental factors. Temperature sensors are the most widespread and are used to measure air, coolant, exhaust, oil, and catalyst temperature. Pressure sensors report on manifold, barometric, or brake line pressure. Position sensors are used in the distributor and measure crankshaft and accelerator throttle position. Fluid flow sensors are employed to monitor fuel consumption or oil circulation. Environmental sensors measure humidity and gas composition and are used in both spark timing and emission control.

Specific problems with each type of sensor vary, but many suffer in environments of extreme temperature and extreme vibration. In addition, most are *costly*. Many sensors require external devices to function properly. Until the reliability and cost factors can be solved, sensors will be the limiting factor in microprocessed engine control.

ADDITIONAL PROBLEMS

The automotive environment provides unique problems for electronic devices. The engine compartment is particularly harsh since temperature ranges from -40C to 120C, humidity from 0.1g/kg to 200g/kg, corrosive and contaminating liquids are present, and vibration, shock, and high electromagnetic impulses all must be considered.

In the Oldsmobile, engineers decided that the best solution was to locate the microprocessor in the passenger compartment. This solves one problem but requires expensive wiring. The Chrysler microprocessor is located in the engine compartment, next to the air cleaner.

A major decision facing the engineers is the trade-off between a general purpose or a specific application microprocessor. The anticipated production volume by the auto makers make the development costs of a specific or dedicated microprocessor cost effective. The Delco/Remy and the TI/RCA units that are currently in development or use do not employ dedicated microprocessors. The primary reason is that these are the first systems to be used in this application, and there has not been the time or the money to develop a custom tailored unit so quickly. Also, these systems are at the forefront of both computer and automotive technology, and the engineers have not clearly defined their needs. In addition, it is not clear that the semiconductor industry could produce a chip to the automotive engineers' specifications.

The problem is further complicated because the technologies and requirements are constantly changing. The automotive manufacturers want workable systems now but expect to continually make changes to reduce cost and increase efficiency. By the mid 1980's nearly every American car will contain one or more microprocessors.[5] Therefore, the potential volume is ten million units per year or more. The microprocessor industry is willing to cater to the needs of the automobile industry, but until more systems are put into use, specific industry needs will remain uncertain.

SOFTWARE AND TESTING CONSIDERATIONS

Development costs fall into three categories: hardware, software, and testing. In addition to the hardware problems, which have already been presented, software presents some unique problems.

Microprocessors have two types of memories, random access memory (RAM), and programmable read only memory (PROM). Instructions consist of a fixed pattern of binary word patterns. Some instructions are permanently introduced into the microprocessor at the time that the chip is manufactured. Other instructions are introduced into the PROM at the final stage of automobile manufacture.

Employing these two methods, a manufacturer might have a microprocessor developed for all of its cars that uses the same basic set of instructions. Later a specific set of instructions could be added that would contain all the specific information relating to the specific engine, model, and options that characterize a particular car. The dealer or service center could also alter some particular instruction in order to correct an existing problem or to upgrade a system in an existing vehicle. All the instructions coded into the PROM are considered the software of the system and, as such, would have to be tested and documented.

If the auto industry were to meet strict standards by mechanical means, maintenance requirements would be more expensive as automobiles would require more servicing to meet the same standards.

The testing function involves more than just testing the microprocessor. It involves insuring that the entire system and the vehicle that it is installed in are working properly.

Government legislation and product liability are also important testing factors. The EPA requires that auto manufacturers provide proof that their vehicles are built to maintain certain levels of emission control for a specific number of miles or years. Future legislation is likely to be more stringent and include areas of safety and economy considerations.

OTHER USES OF MICROPROCESSORS

The use of microprocessors in automobiles is not limited to ignition systems. Other areas under development include: cylinder selection, fuel consumption indicator, fuel injection, and transmission control, all of which can be considered to be emissions and economy considerations. Some of these can be incorporated into an ignition control system. For example, cylinder selection involves varying the number of cylinders employed at any one time to maximize fuel economy at different levels of speed, acceleration, and load.

As for safety, there are many microprocessor applications, some of which are very simple and others highly sophisticated. Among the more interesting include applications to check for low tire pressure, to monitor oil

level and battery charge indicators, to activate airbag actuators, to provide drunk driving prevention, and to monitor radar braking systems. The last two are of particular interest to the federal government. The drunk driving prevention device employs a gas composition sensor mounted in the steering wheel hub which prevents the ignition from working if the alcohol level in the breath of the operator is beyond a certain predetermined level. GM has done research on alcohol interlock systems and has produced devices that have effectively screened between 50% and 75% of drivers with blood alcohol concentrations of 0.1%.[6]

Automatic radar brakes is another application that the government is interested in. These systems automatically apply the automobile's brakes when the radar system detects a potential hazard to close to the front of the vehicle. The limiting factor in such a system is that it is difficult to differentiate between X-Band radar signatures of objects of different sizes and risk potentials. A possible solution is dependent upon the reduced cost of computer memory. If sufficient numbers of radar signatures could be stored, it would be possible to support a high-speed radar hazard analysis and automatically apply the brakes.[7] This system would function only as an emergency crash avoidance capacity activating when a crash is imminent and human interaction has not occurred. Other possible, but not probable, applications include brain wave monitoring devices that would wake up drivers who are falling asleep at the wheel. Also, anxiety and/or aggression interlocks could prevent drivers in an accident-prone frame of mind from being able to start the engine.

There are numerous other applications of microprocessors in automobiles, such as headlights, braking and electronic power control. Digital readout gauges are a logical extension of microprocessed engine control because most of the sensors and digital/analog interface units would be present in automobiles with engine control microprocessors. One problem with digital readout is that they are temperature sensitive, and automobiles are subjected to extreme temperature variation. Digital clocks, radio station indicators, and miles to empty fuel gauges have just become optional on high-priced American cars, and the trend is expected to continue as the costs are reduced.

MAINTENANCE CONCERNS

The use of microprocessors in engine control and other automotive applications will certainly have widespread implications on the reliability and maintenance of the automobile. Currently, auto maintenance is designed to keep the vehicle running; in the future mere running will not be enough. As the purpose of microprocessed engine control systems is to allow a very high level of economy, environmental efficiency, and safety, maintenance will be directed towards maintaining those high standards. This will require high level diagnostic devices and mandatory periodic servicing.

While computer technology will contribute to the high cost and sophistication of auto servicing, it will not be its cause. The reason for the radical change in maintenance is society's insistence on high fuel economy, safety, and low emissions. If the auto industry were to meet strict standards by mechanical means, maintenance requirements would be more expensive as automobiles would require more servicing to meet the same standards.

A lag is likely in educating technicians in local service stations. Until advanced electronics are common and have been around for a while, it is likely that it will be difficult to find the qualified technician to work on your car. It is also likely that blame will be unfairly placed on the microprocessor when the real problem will be that the mechanic is not familiar with the system.

Maintenance on the microprocessor equipped car must be compared with other cars that meet the 1981 pollution standards, not with today's vehicles. An alternative would be for Detroit to have gone another route (i.e., diesel, stratified charge, turbine, etc.) to meet the strict standards rather than to increase the sophistication of an existing but inefficient design. Whether or not these designs could meet the 1981 standards without a great many changes is not clear, but it is likely that they would require less in the way of electronics.

CONCLUSIONS

Microprocessors will definitely be playing an increasingly important role in the automobile. Their use is just beginning and will probably become commonplace as a basic automotive component in the very near future. Because they are so new, many unanswered questions remain about their effect on automobile performance, reliability, maintenance, economy, and perhaps safety. There is no question that microprocessors will make the automobile a more sophisticated piece of machinery, and this will certainly have an impact on maintenance.

While the engine control function of a microprocessor will have little impact on the operator, there are many other applications such as safety, comfort, and convenience that could have a profound affect on the driver and his passengers. Unfortunately, these other applications are not likely to be microprocessed as quickly as engine control. As technology advances, more applications will become feasible. The possibilities are almost endless and will surely make the automobile of the 1980's quite different from the one that we drive today. □

FOOTNOTES

[1]Trevor O. Jones. "Automobile Electronics I: Smaller and Better." *IEEE Spectrum,* (November 1977), p. 34.
[2]E.F. Lindsley. "Buick's Turbocharged V6." *Popular Science,* (September 1977), p. 86.
[3]Bernard M. Oliver. "The Role of Microelectronics in Instrumentation and Control." *Scientific American,* (September 1977), p. 183.
[4]Ronald K. Jugen. "The Automobile: For Better or Worse." *IEEE Spectrum,* (November 1977), p. 32.
[5]"Detroit's New Appetite for Electronic Controls." *Business Week,* (August 28, 1977), p. 64.
[6]Trevor O. Jones. "Some Recent and Future Automotive Electronic Developments." *Science,* (March 18, 1977), p. 1159.
[7]*Ibid.,* p. 1159.

BIBLIOGRAPHY

"Detroit's New Appetite for Electronic Controls." *Business Week,* (August 29, 1977), pp. 64-66.
"Electronics in Motion." *Automotive Industries,* (September 15, 1976), p. 56.
Franson, Paul. "Though Digital ICs Gain Rapidly, Today's Automotive Electronics Remain Mostly Linear, Discrete." *EDN,* (October 20, 1977), pp. 19-20.
Jones, Trevor O. "Automobile Electronics I: Smaller and Better." *IEEE Spectrum,* Vol. 14. (November 1977), pp. 34-35.
Jones, Trevor O. "Some Recent and Future Automotive Electronic Developments." *Science,* Vol. 195. (March 18, 1977), pp. 1156-1160.
Jurgen, Ronald K. "The Automobile: For Better or Worse." *IEEE Spectrum,* Vol. 14. (November 1977), pp. 31-33.
Oliver, Bernard M. "The Role of Microelectronics in Instrumentation and Control." *Scientific American,* (September 1977), pp. 182-190.
Puckett, Gene, J. Marley, J. Gragg. "Automotive Electronics II: The Microprocessor Is In." *IEEE Spectrum,* Vol. 14, (November 1977), pp. 37-39.
Toong, Hoo-Min D. "Microprocessors." *Scientific American,* (September 1977), pp. 146-159.

THE 6800 INVADES THE DARKROOM

By R. B. Lang, P.E. and C. J. Lang

Home computer enthusiasts are constantly looking for interesting applications for their investments. As newcomers to the field of photography, we found the precise temperatures and timing required in developing color prints a bit intimidating. A microcomputer acting as a prompter is a useful addition to the photo lab. Using a real time clock, the computer can not only beep when it is time to do something, it can tell you what to do (via teletype or T.V. typewriter). After the processing is over, the computer will even turn on the lights for you automatically. But if you aren't into photography, keep reading. The hardware and software presented here are applicable to any complex process lasting less than 99 minutes and in which timing is critical. There are many applications in cooking and labwork. You can even use it to wake yourself after that 10 or 15 minute catnap.

HARDWARE

The basic hardware in our system consists of a SWTPC MP6800 computer with 4K, a SWTPC CT-1024 Video Terminal, an AC-30 Cassette Interface, and an MP-L Parallel Interface. The MP-L Interface is necessary for any type of real world interfacing. The SWTPC MP-L Parallel Interface consists of a Motorola 6820 Peripheral Interface Adapter Integrated Circuit (PIA) and the necessary buffering. Sixteen data lines are available for input or output along with 4 control lines. The parallel interface is plugged into I/O slot 0 in the SWTPC computer. Any slot may be used by changing the value of the constant PIA0 in the program.

Not being particularly hardware oriented, we tried to make the additional hardware for the computer prompter as simple as possible. The circuit was developed by experimentation on a breadboard out of spare parts lying around the workbench. The values of resistances and capacitances are not very critical. The values shown in Figure 1 work well.

A microcomputer acting as a prompter is a useful addition to the photo lab. . .it can tell you what to do (via teletype or T.V. typewriter). After the processing is over, the computer will. . .turn out the lights. . .

The operation of the hardware is as follows. A 60 cycle, 12 volt A.C. signal is taken from the computer power supply. The signal is conditioned and fed into the 74C08 CMOS "AND" Gate where it emerges as 60 Hz TTL pulses. The pulses are fed to the 7490 Decade Counter. The output on line 11 of the decade counter is a 6 Hz TTL pulse which is fed to C1 on the SWTPC parallel interface (CB1 Interrupt Input on the 6820 PIA). The peripheral input line CB1 is used to set the interrupt flag of the control register of the PIA six times a second. The interrupts are serviced six times a second by the software.

The rest of the circuit is a beeper that is controlled by the "A" side of the PIA. The input to the beeper is connected to 00 of the SWTPC parallel interface (Line PA0 on the 6820 PIA) and is controlled by the software. Of course, up to 8 output devices such as lights or beepers could be connected to 00-07 or 256 if multiplexed, but at the moment we are only using one beeper. The output signal from the PIA is buffered by passing through the 74C08. The signal is then used to turn the beeper, which

consists of the 555 Timer and its associated components, on and off. We determined that this buffering was necessary to prevent erratic and unreliable operation of the real time clock. An output of a hex byte "01" will turn the beeper on, and an output of "00" will turn it off.

SOFTWARE

In writing the software, our two goals were to make the software as flexible as possible and to keep it as short as possible. The result is a program 235_{10} bytes long with a data area following the program. Data is entered into the program in the format shown in the "comments" section of Program 1. The messages need not be input in their order of output. If an incorrect character is input in the message, it may be deleted by typing a "ctrl 0D'.

The program consists of three main sections. The first section starts with the label "BEGIN" and reads the times and messages to be output. A carriage return and a line feed are output. The minute and second of the first message are read in (assumed to be in decimal) and are converted to hexadecimal representation for storage. Data is stored sequentially beginning at address 010B. A 99_{10} (HEX 63) signals the end of the input and the program jumps to the second section. If a 99_{10} is not read, a hex byte is read after the seconds. This hex byte will later be output on lines 00 through 07 of the Parallel Interface. Next, the message is read one character at a time until it is terminated by a "ctrl D".

The beginning of the second section of the program is marked by the label "INIT" in Program 1. The second section does the outputting of messages at specified times. First, the real time clock is reset to zero. Next, the clock is started by the "CLI", clear interrupt instruction. The program starts to search the data area which begins at address 010A for a byte containing "04". An 04 marks the beginning of a data record. The next byte contains the minute that the message is to be printed or a 63_{16}. A 63_{16} signals that the end of the input has been reached and the program waits until the second changes and then restarts the search at 010A for a message to output. If a 63_{16} is not encountered, the minute and second are compared with the current values of the real time clock minutes and seconds stored in addresses 0020 and 0021. If the times match, the third byte following the 04 is output to the A side of the PIA. This byte controls the beeper circuit. Next, a carriage return and a line feed are output, followed by the minute, second, hex byte and message that are to be printed. The time is converted to decimal representation before output. Next, a zero byte is output to the A side of the PIA to turn off the beeper. The program then waits until the second counter changes, then restarts the search for a message at 010A.

The final section of the program begins with the label "IRQ" and is the real time clock interrupt servicing routine. Six times a second, the clock pulse at C1 on the parallel interface causes bit 7 in control register "B" in the PIA to be set. This in turn causes the IRQ input line to the microprocessor to be grounded. Upon this interrupt signal, the Motorola MIKBUG™ software causes a jump to the interrupt service routine indirectly through the addresses A000 and A001. These addresses have been previously loaded with 00E6, the address of "IRQ". Thus a clock pulse at C1 causes a jump to "IRQ" section of the program. First a check of bit 7 of control register "B" is done to determine that it was the clock that caused the interrupt. Next, bit 7 is cleared. The sixth of a second counter is incremented. If the counter is equal to 6, it is reset to zero and the second counter is incremented. Likewise, if the second counter is equal to 60, it is reset to zero and the minute counter is incremented. A return from interrupt is executed.

Figure 1. Real Time Clock and Beeper Circuit

Circuit labels: C1 OF "B" SIDE OF PARALLEL INTERFACE; +5V; 7490 DECADE COUNTER; 555 TIMER; SPEAKER; 100; .1µF; 1.5K; 10K; +5V; 74C08 CMOS QUAD 2-INPUT AND GATE; 00 OF "A" SIDE OF PARALLEL INTERFACE; +5V; 1N4148; 47K; 27K; 12VAC FROM COMPUTER POWER SUPPLY; 1N4148; .1µF

PROGRAM 1

```
                0002    TITLE '    - 6800 PROMPTER PROGRAM LISTING'
                0003  *     THIS PROGRAM KEEPS TRACK OF REAL TIME AND PROMPTS
                0004  *     THE USER WITH MESSAGES AT SPECIFIED TIMES.
                0005  *     WRITTEN BY R. H. LANG
                0006  * COMMENTS:
                0007  * START PROGRAM BY STORING 0044 AT A048.  TYPE G
                0008  * DATA IS ENTERED IN FORMAT MMSSXXINSTRUCTIONSZ...ETC...99
                0009  * WHERE MM IS THE MINUTE AFTER START THAT MESSAGE IS PRINTED
                0010  *     SS IS THE SECOND AFTER START THAT MESSAGE IS PRINTED
                0011  *     XX IS A HEX BYTE TO BE OUTPUT TO PIA0 WHEN MESSAGE
                0012  *        IS PRINTED
                0013  *     INSTRUCTIONS IS THE MESSAGE (ANY LENGTH) TO BE PRINTED
                0014  *     Z IS THE MESSAGE TERMINATOR (CTRL D)
                0015  *     ...ETC... ARE ANY NUMBER OF MESSAGES.  TIMES DO NOT
                0016  *        HAVE TO BE IN INCREASING ORDER.
                0017  *     99 IS THE INPUT TERMINATOR.
                0018  *     (CTRL 0) TO DELETE PREVIOUS CHARACTER IN MESSAGE
                0019  *
                0020  * AFTER INPUT TERMINATION IS READ, THE CLOCK WILL BE RESET
                0021  * TO ZERO AND STARTED.  THE MESSAGES ARE PRINTED AT THE
                0022  * SPECIFIED TIMES.  BEEPERS, SIRENS AND FLASHING LIGHTS ARE
                0023  * CONNECTED TO PIA0 (A SIDE).
                0024  * TIMER PROGRAM CAN BE RERUN BY STARTING AT LOCATION 0060.
                0025  * CLOCK INPUT IS CONNECTED TO PIA0(B SIDE) CB1
                0026  * **** MIKBUG ENTRY POINTS ***
        E055    0027  BYTE  EQU   X'E055'   READ HEX BYTE INTO A REG
        E0CA    0028  OUT2HS EQU  X'E0CA'   OUTPUT HEX BYTE POINTED TO BY X REG
        E07E    0029  PDATA1 EQU  X'E07E'   OUTPUT CHAR. POINTED TO BY X REG
                0030  *                     UNTIL X'04' ENCOUNTERED
        E0AA    0031  NIBL  EQU   X'E0AA'   READ HEX NIBBLE INTO A REG
        E1AC    0032  INEEE EQU   X'E1AC'   READ CHAR. INTO A REG.
        A000    0033  IR    EQU   X'A000'   IO INTERRUPT POINTER
        8000    0034  PIA0  EQU   X'8000'   PIA ADDRESS
0030            0035        PSELT X'30'
0030            0036  MIN   FCB   X'0'      REAL TIME MINUTES
0020 00         0037  SEC   FCB   X'0'      REAL TIME SECONDS
0021 00         0038  SIXTH FCB   X'0'      REAL TIME 1/6 SECONDS
0022 00         0039  CSEC  FCB   X'0'      CURRENT SECONDS
0023 00         0040  CR    FCB   X'0D'     CARRIAGE RETURN
0024 0D         0041  LF    FCB   X'0A'     LINE FEED
0025 0A         0042  EOT   FCB   X'04'     EOT
0026 04         0043  XSAVE FDB   X'0000'   RESERVE AREA TO SAVE X REG
0027 0000       0044  DECI  JSR   NIBL      CONVERT DECIMAL TO HEX FOR STORAGE
0029 BD
002A E0AA       0045        INX
002C 08         0046        ASLA
002D 48         0047        TAB
002E 16         0048        ASLA
002F 48         0049        ASLA
0030 48         0050        ABA
0031 1B         0051        TAB
0032 16         0052        JSR   NIBL
0033 BD
0034 E0AA       0053        ABA
0036 1B         0054        TAB
0037 16         0055        RTS
0038 39
0039 DF         0056  CRLF  STX   XSAVE     SAVE X REG
003A 27
              0057        LDX   .CR       POINT X REG TO CR,LF,EOT
003B CE
003C 0024       0058        JSR   PDATA1    OUTPUT DATA POINTED TO BY X REG
003E BD
003F E07E       0059        LDX   XSAVE     RESTORE X REG
0041 DE
0042 27         0060        RTS
0043 39         0061  BEGIN LDX   .DATA     INITIALIZE X REG TO START OF DATA AREA
```

```
0044 CE
0045 010A
0047 BD         0062  STO4  BSR   CRLF      SUBROUTINE WILL OUTPUT CR/LF
0048 F0
0049 BD         0063  MORE  BSR   DECI      READ MINUTES
004A DE
004B A7         0064        STAA  0,X       STORE MINUTES
004C 00
              0065        CMPA  .99       IF MIN=99 THEN END
004D 81
004E 63
004F 27         0066        BEQ   INIT      OF INPUT,BRANCH TO CLOCK
0050 1C
0051 BD         0067        BSR   DECI      READ SECONDS
0052 D6
              0068        STAA  0,X       STORE SECONDS
0053 A7
0054 00
              0069        JSR   BYTE      READ BYTE TO BE OUTPUT ON PIA0
0055 BD
0056 E055
0058 08         0070        INX
              0071        STAA  X         STORE BYTE IN DATA AREA
0059 A7
005A 00
005B BD         0072  READ  JSR   INEEE     READ PROMPTING MESSAGE
005C E1AC
005E 08         0073        INX
              0074        STAA  0,X       STORE MESSAGE
005F A7
0060 00
              0075        CMPA  .X'0F'    CHECK FOR DELETE 'CTRL 0'
0061 81
0062 0F
0063 26         0076        BNE   NEXT
0064 02
0065 09         0077        DEX
0066 09         0078        DEX
              0079  NEXT  CMPA  .X'04'    IF EOT, THEN END OF MESSAGE
0067 81
0068 04
0069 27         0080        BEQ   STO4
006A DC
006B 20         0081        BRA   READ      READ MORE MESSAGE
006C EE
006D 0F         0082  INIT  SEI
              0083        LDX   .IRQ      STORE ADDRESS OF CLOCK INTERRUPT
006E CE
006F 00E6
0071 FF         0084        STX   IR
0072 A000
0074 4F         0085        CLRA
              0086        STAA  PIA0+1
0075 B7
0076 8001
              0087        STAA  PIA0
0078 B7
0079 8000
007A 86         0088        LDAA  .7
007B 07
007C B7         0089        STAA  PIA0+3    CLOCK AT I/O PORT #0
007D 8003
007F 86         0090        LDAA  .0        RESET CLOCK TO ZERO
0080 00
0081 97         0091        STAA  MIN
0082 20
0083 97         0092        STAA  SEC
0084 21
0085 97         0093        STAA  SIXTH
0086 20
0087 22
0088 0E         0094        CLI             START CLOCK
              0095  REDO  LDX   .DATA
0089 CE
008A 010A
```

Figure 2. Sample Input for Cibachrome Processing

PHOTOPROCESSING EXAMPLE

The system presented in this article is a general purpose timer and applicable to any short time process. We have found the prompter system very useful in doing photographic print processing using the Cibachrome* process.

To run the program, begin by storing 0044 in locations A048 and A049. Type G to begin. Figure 2 is a listing of the input for the 12 minute Cibachrome developing procedure. Figure 3 shows a portion of memory after enter-

ADDRESS	CONTENTS																
010A	04	00	00	01	2A	2A	43	49	42	41	43	48	52	4F	4D	45	
		00	00	01	*	*	C	I	B	A	C	H	R	O	M	E	
011A	2A	2A	04	00	02	00	53	54	41	51	54	20	42	59	20	50	
	*	*		00	02	00	S	T	A	R	T		B	Y		P	
. . .	...																
0278	20	48	41	49	52	20	44	52	59	45	52	04	13	14	01	50	
		H	A	I	R		D	R	Y	E	R		19	20	01	P	
0288	52	49	4E	54	20	46	49	4E	49	53	48	45	44	21	04	63	
	R	I	N	T		F	I	N	I	S	H	E	D	!		99	

Figure 3. Hexadecimal Representation of Messages in Memory with Decimal or ASCII Equivalent Shown Below

ing the input in Figure 2. To restart the program to process additional prints, just hit the reset button, load 006D in addresses A048 and A049 and type G. To save the Cibachrome instructions and the program on tape, set A002 to 00, A003 to 20, A004 to 2, and A005 to 97. Use the MIKBUG "P" command to save the program and data on tape. The program and data can then be loaded from tape using the MIKBUG "L" command and run using the restart instructions. □

Photograph on Page 82 by Shelley Wright

*Cibachrome is a simple process developed by Ilford for making color prints from slides. The "Discovery Kit", for less than $20, contains all the supplies needed to produce 20 5" x 4" prints from your favorite slides. The Kit comes complete with an ingenious developing tank which allows the developing to proceed in a lighted room. The tank contains a holding cup which holds the chemical until the tank is turned on its side, at which time the chemical flows over the print. When the tank is uprighted, the spent chemical flows out a bottom drain, and the next chemical can be added to the holding cup.

```
008L 20      0096        BRA   ADD
008L 02
008E 08      0097 NOTMIN INX
008F 08      0098 NOTSEC INX
             0099 ADD    LDAA  X
0090 A6
0091 00
0092 08      0100        INX
             0101        CMPA  #X'04'  CHECK FOR BEGINNING OF RECORD
0093 81
0094 04
0095 26      0102        BNE   ADD
0096 F9
             0103        LDAA  X
0097 A6
0098 00
             0104        CMPA  #99     CHECK FOR END OF DATA
0099 81
009A 63
009B 27      0105        BEQ   DEL
009C 27
009C 08      0106        INX
             0107        CMPA  MIN     CHECK MIN OF ENTRY AGAINST CURRENT MIN
009E 91
009F 20
00A0 26      0108        BNE   NOTMIN
00A1 EC
             0109        LDAA  X
00A2 A6
00A3 00
00A4 08      0110        INX
             0111        CMPA  SEC     CHECK SEC OF ENTRY AGAINST CURRENT SEC
00A5 91
00A6 21
00A7 26      0112        BNE   NOTSEC
00A8 E6
             0113        LDAA  X       LOAD BYTE TO SEND TO PIAO
00A9 A6
00AA 00
             0114        LDAB  SEC     SAVE CURRENT SEC COUNT
00AB D6
00AC 21
             0115        STAB  CSEC
00AD 07
00AE 23
             0116        STAA  PIAO    SEND BYTE TO PIAO
00AF B7
00B0 8000
00B2 8D      0117        BSR   CRLF    SUBROUTINE TO OUTPUT CR/LF
00B3 85
00B4 09      0118        DEX           BACK TO SECONDS
00B5 09      0119        DEX           BACK TO MINUTES
00B6 8D      0120        BSR   TIMEP   WRITE OUT MIN
00B7 14
00B8 8D      0121        BSR   TIMEP   WRITE OUT SEC
00B9 12
             0122        JSR   OUT2HS  WRITE OUT DATA SENT TO PIAO
00BA BD
00BB E0CA
             0123        JSR   PDATA1  WRITE OUT MESSAGE
00BE BD
00BF E07E
00C0 4F      0124        CLRA
             0125        STAA  PIAO    SEND ZERO BYTE TO PIAO
00C1 B7
00C2 8000
             0126 DEL    LDAB  CSEC
00C4 D6
00C5 23
             0127 DELAY  CMPB  SEC     WAIT TILL SEC CHANGES
00C6 D1
00C7 21
00C8 26      0128        BNE   PEGU
00C9 8F
00CA 20      0129        BRA   DELAY
00CB FA
             0130 TIMEP  LDAB  X       OUTPUT TIME IN BASE 10
00CC E6
00CD 00
00CE 06      0131        INX
             0132        STX   XSAVE   SAVE X REG
00CF DF
00D0 27
00D1 4F      0133        CLRA          BCD CONVERSION FOR OUTPUT OF TIME
             0134 N1     CMPB  #10
00D2 C1
00D3 0A
00D4 2B      0135        BMI   N2
00D5 66
             0136        ADDA  #X'10'
00D6 8B
00D7 10
             0137        SUBB  #10
00D8 C0
00D9 0A
00DA 20      0138        BRA   N1
00DB F6
00DC 1B      0139 N2     ABA
00DD 36      0140        PSHA
00DE 30      0141        TSX
             0142        JSR   OUT2HS
00DF BD
00E0 E0CA
00E2 31      0143        INS
             0144        LDX   XSAVE   RESTORE X REG
00E3 DE
00E4 27
00E5 39      0145        RTS
             0146 IRQ    LDAA  PIAO+3  BEGINNING OF CLOCK INTERRUPT ROUTINE
00E6 B6
00E7 8003
00E9 2A      0147        BPL   EXIT    EXIT IF INTERRUPT FLAG IS CLEAR
00EA 1E      0148        LDAA  PIAO+2  CLEAR INTERRUPT FLAG
00EB B6
00EC 8002
             0149        INC   SIXTH   INCREMENT 1/6 SECOND COUNTER
00EE 7C
00EF 0022
             0150        LDAA  #6
00F1 86
00F2 06
             0151        CMPA  SIXTH
00F3 91
00F4 22
00F5 26      0152        BNE   EXIT    IF SIXTH.NE.6 THEN EXIT
00F6 12
             0153        CLR   SIXTH   IF SIXTH.EQ.6 THEN RESET SIXTH TO ZERO
00F7 7F
00F8 0022
             0154        INC   SEC     AND INCREMENT SECONDS
00FA 7C
00FB 0021
             0155        LDAA  #60
00FC 86
00FD 3C
             0156        CMPA  SEC
00FF 91
0100 21
0101 26      0157        BNE   EXIT    IF SEC.NE.60 THEN EXIT
0102 06
             0158        CLR   SEC     IF SEC.EQ.60 THEN RESET SEC TO ZERO
0103 7F
0104 0021
             0159        INC   MIN     AND INCREMENT MINUTE
0106 7C
0107 0020
0109 3B      0160 EXIT   RTI           END OF CLOCK INTERRUPT
             0161 DATA   FCB   X'04'
010A 04
             0162        END
```

Computer Generated Morse Code

By Jim McClure

For some time I have been interested in becoming a radio amateur and obtaining a General Class license. Unfortunately, the test administered by the Federal Communications Commission requires that a General Class radio amateur be able to send and receive Morse Code at a minimum speed of thirteen words per minute, not a natural ability by any means. I was reminded that "practice makes perfect" applied doubly to code, and the only way to learn it was to copy code sent by a proficient ham until I had mastered the required speed. Finding a proficient ham required me to look no farther than an Altair 8800 microcomputer.

The accompanying program, written for an 8080 microprocessor, accepts text from a console device and outputs the text to an audio amplifier as Morse Code. Since I am not a hardware type, the program was written to perform all necessary tone generation and modulation to simulate a code practice oscillator. All that is needed in the way of special hardware, beside the computer and a terminal, is a low-fi audio amplifier. While this program was intended for an 8080, it can be rewritten for other systems using the flowcharts presented with this article.

SPEED	DASHL Sets length of dash	DOTL Sets length of dot	SPACEL Sets delay between units	PAUSEL Sets delay between characters
5 WPM	80H	30H	0AH	0FFFFH
10 WPM	40H	10H	08H	070FFH
15 WPM	30H	0CH	06H	040FFH

Table 1.

When activated, the program prints a question mark on the console and waits for a line of text to be typed. (The rubout key will delete a previously entered character if a mistake is made.) After a carriage return is received, conversion from ASCII to Morse Code begins.

In addition to all upper case letters, the symbols for period, comma, semicolon, colon, and question mark are accepted by the program. Any other characters are ignored.

Conversion from ASCII to Morse Code is accomplished through a master table which contains an entry for each legal character. Each entry consists of two bytes. The first byte indicates the number of sending units (dashes or dots) for the desired character. Each bit of the second byte, read from right to left, represents a sending unit. If the bit is a zero, the corresponding sending unit will be a dot. Otherwise a dash will be sent. Figure 1 gives an example.

Table entry: Byte 1 = 00000010B   Byte 2 = 00000010B

According to byte 1, we read two bits of byte 2 from right to left, generating a dot for the zero bit and a dash for the next bit since it is a one. This gives us the Morse Code equivalent of the letter 'A'.

Figure 1.

The tone generation portion of the program operates on the same principle as many of the popular no-hardware computer music synthesizers. The processor turns an output line on and off at a high rate, thereby generating a square wave of a frequency in the audio range. This line is then coupled to an amplifier, where the wave's level is boosted to drive a speaker. The output line is usually one bit of a parallel interface. However, all of my interfaces are serial type. Having no desire to buy or build a parallel port, I decided that the input to the audio

amplifier could be connected to the 'INTE' LED on the front panel of the computer by means of a .22 microfarad capacitor. This LED is lit whenever interrupts are enabled by the processor. This means that the LED can be turned on by executing an 'EI' (enable interrupts) instruction and turned off by executing a 'DI' (disable interrupts) instruction. Instead of modulating an output port, the program simply executes 'EI' and 'DI' instructions at a fast rate, causing the LED to blink, and thereby producing a tone for the amplifier. Of course, if a latching parallel interface is available, all this can be discarded by following the procedure detailed in Figure 2.

Replace all 'EI' instructions with:
MVI A,1
OUT address of parallel port

Replace all 'DI' instructions with:
MVI A,0
OUT address of parallel port

Connect audio input line in series with a .22 microfarad capacitor to bit 0 of your parallel port.

Figure 2.

All timing is controlled by four variables which are set at the beginning of the program to send code at about five words per minute. As receiving speed increases, the values of these variables may be adjusted to send at a higher rate. Table 1 lists the four variables and values for common speeds.

In order to run the program, two routines must be added which input from and output to the console device. The addresses of these two routines must be stored in the main program as follows:

0206H — Store low address of console input routine
0207H — Store high address of console input routine
0209H — Store low address of console output routine
020AH — Store high address of console output routine

The input routine should return a character from the console in register A. The output routine should print a character from register C. No other registers are to be modified.

One last address should be added at locations 020CH (low byte) and 020DH (high byte). This is the address that the program jumps to when a carriage return with no text is typed in response to the question mark prompt. This address should be the entry point of whatever resident monitor is present in the system.

With regard to the hardware connection between the audio amplifier and the computer, the input cable (should be shielded) may be attached to the Display Control board of an Altair 8080A just behind the 'INTE' light and run out the back of the unit. If no front panel is available, a connection may be made to the S-100 bus at pin 28. Make sure that the amplifier being used has a fairly high input impedance so as not to load down any internal computer circuits, and don't forget the .22 microfarad capacitor between the computer and the amplifier. A good ground connection is also important to minimize hum and noise pickup.

If a parallel port is being used, the connection to the amplifier should be made according to the instructions given in Figure 2.

While all of this may seem like a lot of work, it goes quickly and yields surprisingly good results. You'll look far and wide before you find a ham who sends as smoothly as the computer. □

Flowchart 1. Main Routine

Flowchart 2. Input Subroutine

Flowchart 3. Xmit Subroutine

Flowchart 4. Timing and Unit Transmission Subroutines

PROGRAM LISTING

```
0080 =            BUF      EQU   80H        ;TERMINAL READ BUFFER
0020 =            SPACE    EQU   20H        ;SPACE CHARACTER
000D =            CR       EQU   0DH        ;CARRIAGE RETURN
000A =            LF       EQU   0AH        ;LINE FEED
0007 =            BELL     EQU   07H        ;BELL
007F =            RUB      EQU   7FH        ;RUBOUT
00D0 =            PITCH    EQU   0D0H       ;SETS FREQUENCY OF TONE
0080 =            DASHL    EQU   80H        ;SETS DELAY FOR DASH
0030 =            DOTL     EQU   30H        ;SETS DELAY FOR DOT
000A =            SPACEL   EQU   0AH        ;SET DELAY BETWEEN SEND UNITS
FFFF =            PAUSEL   EQU   0FFFFH     ;SET DELAY BETWEEN CHARACTERS
                  ;
                  SENDB:
0100 CDB901       CALL     INPUT ;READ A LINE FROM CONSOLE
0103 110000       LXI      D,BUF
0106 B7           ORA      A     ;CHECK COUNT OF CHARACTERS
0107 CA0E02       JZ       DONE  ;IF ZERO THEN QUIT
010A 47           MOV      B,A   ;COUNT OF CHARACTERS INTO REG. B
                  SENDL:
010B 1A           LDAX     D     ;GET CHARACTER TO BE SENT
010C 4F           MOV      C,A   ;SAVE IN REG. C
010D FE20         CPI      SPACE
010F CC3C01       CZ       PAUSE ;IF SPACE THEN PAUSE
0112 CC3C01       CZ       PAUSE
0115 CA3401       JZ       SENDN ;AND SEND NEXT CHARACTER
0118 FE28         CPI      40
011A DA3401       JC       SENDN ;MAKE SURE IT IS VALID
011D FE5B         CPI      'Z'+1 ;ELSE SKIP IT
011F D23401       JNC      SENDN
0122 211402       LXI      H,TABLE ;SET POINTER TO CONVERSION TABLE
0125 D5           PUSH     D     ;SAVE REG. D-E
0126 D628         SUI      40    ;SUBTRACT LOWEST VALID CHARACTER
0128 17           RAL            ;TIMES 2
0129 5F           MOV      E,A   ;SET UP ADDITION
012A 1600         MVI      D,0   ;CLEAR D REG.
012C 19           DAD      D     ;POINT TO TABLE ENTRY
012D CD5201       CALL     XMIT  ;SEND THE CHARACTER
0130 D1           POP      D     ;RESTORE POINTER TO BUFFER
0131 CD3C01       CALL     PAUSE ;WAIT BEFORE NEXT CHARACTER
                  SENDN:
0134 13           INX      D     ;NEXT CHARACTER
0135 05           DCR      B
0136 C20B01       JNZ      SENDL ;LOOP TIL BUFFER EMPTY
0139 C30001       JMP      SENDB ;IF EMPTY START OVER
                  ;
                  PAUSE:
013C F5           PUSH     PSW   ;SAVE STATUS
013D D5           PUSH     D
013E 1E02         MVI      E,2
                  PAU1:
0140 21FFFF       LXI      H,PAUSEL
0143 2D           DCR      L
0144 C24301       JNZ      $-1   ;DELAY BETWEEN CHARACTERS
0147 25           DCR      H
0148 C24301       JNZ      $-5
014B 1D           DCR      E
014C C24001       JNZ      PAU1
014F D1           POP      D
0150 F1           POP      PSW   ;RESTORE STATUS
0151 C9           RET
                  ;
                  XMIT:
0152 F5           PUSH     PSW   ;SAVE STATUS
0153 C5           PUSH     B
0154 46           MOV      B,M   ;GET NUMBER OF UNITS
0155 AF           XRA      A     ;CLEAR ACCUMULATOR
0156 B0           ORA      B     ;SEE IF REG. B IS ZERO
0157 CA6D01       JZ       XMITE ;IF SO THEN DO NOT SEND
015A 23           INX      H
015B 4E           MOV      C,M   ;SAVE BYTE CODE IN REG. C
                  XMITL:
015C 79           MOV      A,C   ;GET BYTE CODE FROM REG. C
015D B7           ORA      A     ;CLEAR CARRY FLAG
015E 1F           RAR            ;TEST FIRST BIT
015F DC7001       CC       DASH  ;IF HIGH THEN SEND A DASH
0162 D48901       CNC      DOT   ;IF LOW THEN SEND A DOT
0165 4F           MOV      C,A   ;SAVE SHIFTED CODE
0166 CDA201       CALL     SPACER ;DELAY BEFORE SENDING NEXT UNIT
0169 05           DCR      B     ;CHECK COUNT
016A C25C01       JNZ      XMITL ;LOOP UNTIL ZERO
                  XMITE:
016D C1           POP      B     ;RESTORE ALL REGS.
016E F1           POP      PSW
016F C9           RET
                  ;
                  DASH:
0170 F5           PUSH     PSW   ;SAVE ALL REGS.
0171 E5           PUSH     H
0172 2680         MVI      H,DASHL ;DELAY FOR DASH
                  DASH1:
0174 F3           DI
0175 2ED0         MVI      L,PITCH ;FREQUENCY DETERMINING
0177 2D           DCR      L
0178 C27701       JNZ      $-1
017B FB           EI              ;NEXT CYCLE
017C 2ED0         MVI      L,PITCH ;MAKE SQUARE WAVE
017E 2D           DCR      L
017F C27E01       JNZ      $-1
0182 25           DCR      H      ;MAJOR LOOP
0183 C27401       JNZ      DASH1
0186 E1           POP      H
0187 F1           POP      PSW    ;RESTORE REGS.
0188 C9           RET
                  ;
                  DOT:
0189 F5           PUSH     PSW   ;SAVE ALL REGS.
018A E5           PUSH     H
018B 2630         MVI      H,DOTL ;DELAY FOR DOT
                  DOT1:
018D F3           DI
018E 2ED0         MVI      L,PITCH ;FREQUENCY DETERMINING
0190 2D           DCR      L
0191 C29001       JNZ      $-1
0194 FB           EI
0195 2ED0         MVI      L,PITCH ;MAKE SQUARE WAVE
0197 2D           DCR      L
0198 C29701       JNZ      $-1
019B 25           DCR      H      ;MAJOR LOOP
019C C28D01       JNZ      DOT1
019F E1           POP      H
01A0 F1           POP      PSW    ;RESTORE REGS.
01A1 C9           RET
                  ;
                  SPACER:
01A2 F5           PUSH     PSW   ;SAVE ALL REGS.
01A3 E5           PUSH     H
01A4 260A         MVI      H,SPACEL ;PAUSE FOR A DOT
                  SPAC1:
```

```
01A6 2ED0         MVI      L,PITCH ;FREQUENCY DETERMINING
01A8 2D           DCR      L
01A9 C2A801       JNZ      $-1
01AC 2ED0         MVI      L,PITCH ;FREQUENCY DETERMINING
01AE 2D           DCR      L
01AF C2AE01       JNZ      $-1
01B2 25           DCR      H
01B3 C2A601       JNZ      SPAC1
01B6 E1           POP      H      ;RESTORE REGS.
01B7 F1           POP      PSW
01B8 C9           RET
                  ;
                  ; BUFFERED INPUT ROUTINE
                  ; TYPING A RUBOUT WILL DELETE THE PREVIOUSLY TYPED CHARACTER
                  ; AND RE-ECHO IT. TYPING A RUBOUT WITH NO CHARATERS ON LINE
                  ; WILL ECHO A BELL. A BELL WILL ALSO BE ECHOED IF THE MAXIMUM
                  ; LINE LIMIT OF 80 CHARACTERS IS EXCEEDED
                  ;
                  INPUT:
01B9 CDFA01       CALL     CRLF  ;ECHO A CR AND A LF
01BC 0E3F         MVI      C,'?' ;DISPLAY PROMPT
01BE CD0802       CALL     COUT
01C1 210080       LXI      H,BUF ;INPUT BUFFER
01C4 0650         MVI      B,80  ;LINE LIMIT
                  IN1:
01C6 CD0502       CALL     CINP  ;GET CHARACTER FROM CONSOLE
01C9 77           MOV      M,A   ;STORE IN MEMORY
01CA FE0D         CPI      CR    ;SEE IF FINISHED
01CC CAEA01       JZ       INPEND
01CF FE7F         CPI      RUB   ;SEE IF A DELETE
01D1 CAEE01       JZ       RUBCHR
01D4 23           INX      H     ;NEXT LOCATION
01D5 05           DCR      B     ;CHECK COUNT
01D6 C2E301       JNZ      IN2
01D9 04           INR      B     ;IF LIMIT EXCEEDED
01DA 2B           DCX      H
                  IERROR:
01DB 0E07         MVI      C,BELL ;ECHO A BELL
01DD CD0802       CALL     COUT
01E0 C3C601       JMP      IN1   ;AND GET NEXT CHARACTER
                  IN2:
01E3 4F           MOV      C,A
01E4 CD0802       CALL     COUT  ;ECHO CHARACTER
01E7 C3C601       JMP      IN1
                  INPEND:
01EA 3E50         MVI      A,80  ;RETURN WITH NUMBER OF
01EC 90           SUB      B     ;CHARACTERS IN REG. A
01ED C9           RET
                  RUBCHR:
01EE 78           MOV      A,B   ;CHECK COUNT
01EF FE50         CPI      80
01F1 CADB01       JZ       IERROR ;IF ZERO THEN AN ERROR
01F4 04           INR      B      ;ELSE
01F5 2B           DCX      H      ;DELETE LAST CHARACTER
01F6 7E           MOV      A,M
01F7 C3E301       JMP      IN2    ;AND RE-ECHO IT
                  ;
                  CRLF:
01FA 0E0D         MVI      C,CR   ;ECHO A CARRIAGE RETURN
01FC CD0802       CALL     COUT
01FF 0E0A         MVI      C,LF   ;AND A LINE FEED
0201 CD0802       CALL     COUT
0204 C9           RET
                  ;
                  ; JUMPS TO I/O ROUTINES
                  ;
0205 C30000       CINP:    JMP   00000H     ;CONSOLE INPUT ROUTINE
0208 C30000       COUT:    JMP   00000H     ;CONSOLE OUTPUT ROUTINE
020B C30000       BOOT:    JMP   00000H     ;MONITOR ENTRY POINT
                  ;
                  DONE:
020E CDFA01       CALL     CRLF  ;PRINT A CR AND LF
0211 C30B02       JMP      BOOT  ;RETURN TO SYSTEM MONITOR
                  ;
                  ; TABLE OF LETTERS
                  ;
                  TABLE:
0214 062D         DB       6,101101B
0216 062D         DB       6,101101B
0218 0000         DB       0,0B
021A 0000         DB       0,0B
021C 0633         DB       6,110011B
021E 0000         DB       0,0B
0220 062A         DB       6,101010B
0222 0509         DB       5,01001B
0224 051F         DB       5,11111B
0226 051E         DB       5,11110B
0228 051C         DB       5,11100B
022A 0518         DB       5,11000B
022C 0510         DB       5,10000B
022E 0500         DB       5,00000B
0230 0501         DB       5,00001B
0232 0503         DB       5,00011B
0234 0507         DB       5,00111B
0236 050F         DB       5,01111B
0238 0607         DB       6,000111B
023A 0615         DB       6,010101B
023C 0000         DB       0,0B
023E 0000         DB       0,0B
0240 0000         DB       0,0B
0242 060C         DB       6,001100B
0244 0000         DB       0,0B
0246 0202         DB       2,10B
0248 0401         DB       4,0001B
024A 0405         DB       4,0101B
024C 0301         DB       3,001B
024E 0100         DB       1,0B
0250 0404         DB       4,0100B
0252 0303         DB       3,011B
0254 0400         DB       4,0000B
0256 0200         DB       2,00B
0258 040E         DB       4,1110B
025A 0305         DB       3,101B
025C 0402         DB       4,0010B
025E 0203         DB       2,11B
0260 0201         DB       2,01B
0262 0307         DB       3,111B
0264 0406         DB       4,0110B
0266 040B         DB       4,1011B
0268 0302         DB       3,010B
026A 0300         DB       3,000B
026C 0101         DB       1,1B
026E 0304         DB       3,100B
0270 0408         DB       4,1000B
0272 0306         DB       3,110B
0274 0409         DB       4,1001B
0276 040D         DB       4,1101B
0278 0403         DB       4,0011B
```

BUSINESS EDITORIAL

By Rodnay Zaks
SYBEX, Inc.

BUSINESS MICROCOMPUTERS: FRAUD OR REALITY?

Microcomputers have been widely advertised as being applicable to many type of small businesses. Within the last several months hardware costs have dropped below the $10,000 mark, thus putting the prospect of automation closer to the small businessman's pocketbook.

With this decrease in cost has come, surprisingly enough, an extremely high degree of capability, or more correctly, probable capabilities. The essential question, however, is: do microcomputers offer total capabilities to the businessman today?

The answer to this question is an emphatic No! But to understand why, an understanding of where the business micro came from and what it is expected to do must be explored.

FROM HOME TO BUSINESS COMPUTERS

An article in the January 1975 issue of Popular Electronics, by Leslie Solomon, revealed the existence of a low cost microcomputer available to hobbyists. The computer was, of course, the MITS Altair, based on the 8080 microprocessor. With the publication of the article came the beginning of what was projected to be a huge home computing market.

Within months, small companies were forming to fill the hardware gap, and no end appeared in sight. Yet three years later the hobbyist market has bottomed out and the industry is targeting to an even more promising market: the small business. However, with this market turnaround has come different problems for the manufacturers.

Businessmen cannot and will not tolerate the lengthy delays that so characterized the industry in its hobbyist days. Reliability has become an even more important factor; and, of course, cost.

Taking all of these problems into account, the industry has done well in providing solutions to meet delivery dates, and improving industry to end user relations.

But with improving the hardware and reducing costs, the microcomputer manufacturer has created yet another problem: that of representing current systems as the businessman's rosetta stone — the cure-all that will solve all the ills and management problems of ALL small businesses in the country. Is this a Fraud or a Reality?

REQUIREMENTS OF BUSINESS COMPUTING

Automating any small business requires the availability of specialized files and file management programs for: accounts receivable, accounts payable, payroll, general ledger, inventory, tax, bank accounts, sales reports, and other reports, journals or ledgers that are important to a specific business type. To automate these types of activities offers a welcome benefit when time is of consideration. The results should be threefold: improved accuracy, almost instant availability of reports and statistics and a reduction in manpower.

However, to achieve all these benefits a method of file management must be established. This means that whenever a transaction is performed, all necessary programs or subsystems must be properly updated without performing extra entries of the same information.

For example, when a sales transaction is entered the customer and sales information are entered along with a billing date. On entry, the accounts receivable journal should be updated along with establishing a new record to the customer file. While this is taking place, or in sequence with, the inventory records are updated to reflect the notation that an item(s) have sold and are physically removed from the shelves. When the billing date is established, a shipping date may be in order which causes another sequence of events to take place.

Sounds complicated, but is only a direct reflection of what is done every day under manual methods.

Another requirement for business computer systems is that the differences between businesses must be taken into consideration. An accounting system that works well for a hardware store will probably be of no use to a book dealer or dry cleaning operation. Each business has different needs; even those businesses engaged in the same type of activity. Consequently, both the hardware — physical computer system — and software — the working programs — must be designed to fit the defined user's needs.

Flexibility is also essential in a business environment. Initially the needs of the business might be served by a number of simple software packages performing the traditional functions. However, it might quickly become desirable to add other customized routines to this set. Unless the competence exists in-house and all packages being utilized are fully documented, the task necessary to add the required additional facilities might become prohibitive.

The requirements of the small business are technically best served by a highly complex set of programs customized for the specific business. Clearly, this approach is not now realistic in view of the general unavailability of sophisticated software and the very high cost of programming relative to the cost of the hardware. Limitations in the value of the business programs will therefore exist.

THE HARDWARE

Every microcomputer system first requires a box containing the microcomputer itself, i.e. the microprocessor board, the memory boards, any required interface boards, plus a power supply. In addition, the system requires a business quality printer, a CRT terminal and some form of mass storage.

The microcomputer itself often appears as the crucial choice in the selection of a business system. It is probably the least important one. The speed of the microprocessor itself is almost irrelevant. Because nearly all business systems are implemented in a high level language, the efficiency of the software interpreter or compiler which is used to execute this high level language is the item of crucial importance for the efficiency of the system.

There are naturally advantages and disadvantages inherent to each microprocessor. For example, in order to enjoy the possible benefits of standardized boards, any system providing an S-100 bus offers an advantage. It requires, in turn, an 8080 or Z80 microprocessor. However, provided that the sufficient set of peripherals be available from the start, the option to be able to add new fancy boards may be more appealing to the hobbyist than the business person, and other busses than S-100 might be equaly acceptable. The choice of this beautiful microprocessor box may therefore be based on the established reputa-

tion of the manufacturer, its assumed reliability, or the possible advantages of its bus structure.

The hardware items which may have the most important significance for the businessman are by far the peripherals. It must be remembered that the cost of the peripherals will usually be the dominant cost in a system. Peripherals are likely to be usable over a significant period of time, whereas the microcomputer mainframe is likely to be obsolete in a short amount of time. It might be more valuable to invest time in the correct selection of the long lived expensive peripherals than in the selection of the mainframe.

THE SOFTWARE

Software refers to all the programs necessary to make efficient use of the set of hardware resources available on a system. *At this time, no complete business software facility exists for microcomputers!*

Partial implementations exist and a number of simple packages are now available which will perform (usually separately) payroll, accounts receivable, general ledger, and other functions. However, the crucial task of simultaneous file management and sequential activation of selected programs is, as yet, not implemented. Such software solves business problems individually but does not provide the comprehensive facility needed for the efficient use of the hardware resources.

Because good comprehensive software is not yet available, microcomputers do not have the capability of solving all the business problems that are advertised.

IS THIS A FRAUD?

Current software available for microcomputers makes them capable of solving a large number of tasks commonly associated with business accounting and bookkeeping. Because of the limitation in the automatic file handling capability of most of these programs, the computerization of these tasks may not result in any savings in terms of personnel. The entry of data for computer use tends to be longer and more complex than the manual typing of invoices or filling out of conventional forms. This is because a number of extra fields are required, and the entry format is highly structured. As a result, in most small companies computerization might require somewhat more manpower than less.

In addition, the possible unreliability of hardware and software components might result in catastrophic system breakdown. Every small busi-

ness owner will fully realize the computer "down" at the time that payroll checks should be generated, especially when the data needed has been saved on a single disk file which has just been wiped because of "accidental" error. These drawbacks are real.

THE REAL ADVANTAGES

The real value of contemporary microcomputer systems, with their limited software, lies in two areas: management education, and future savings.

Every user of new and complex machinery must spend a significant period of time to learn the skills necessary to evaluate and control it. Therefore, it is considered highly advisable to practice on a used computer, rather than the expensive new one, the first time around.

With the introduction of computers in a business, a phenomenon known as computer shock occurs. The radical change of procedures required by computer programs often causes personnel to leave, rebel, or otherwise lose their efficiency. Similarly, catastrophic initial failures are likely to occur in the form of data being wiped out or not being produced at the right time.

However, because of the limited cost of microcomputers today, a heretofore unknown opportunity exists for the business owner to familiarize himself and his employees at minimal cost with this new technique.

In summary, microcomputers today offer the capability to learn business computerization at a modest cost. In addition, they have the potential in specific situations to bring modest or sometimes significant savings in the case of business expansion. Finally, they may be able to supply business capabilities which were simply not existent before.

For these reasons, current microcomputers are likely to pay for themselves several times over in direct business benefits as well as education for the business owner. They are far from having attained the true business automation capabilities which larger computers have demonstrated so far and should not be presented as such. Business microcomputers are a reality. The realistic evaluation of their limitations is also a necessity. □

This page is reserved for any company editorial geared to the small business market. Please send your editorial directly to Carl Warren, Senior Editor, INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701.

Most literature in the computer field today is geared towards people who have a great deal of familiarity with computers. The literature is not geared towards people who are business oriented. Most of the applications for computers in the coming years are going to be for the businessman. the purpose of this article is to help bridge the gap, and to make computers and computer applications understandable to the businessman.

We will discuss what use the computer has to a business and will show how a computer fits into an overall business operation. Figure 1 shows the fundamental concept of what a computer does. The input to the computer is data. Examples of data would be: Joe Dokes paid $10 for a dozen golf balls this morning, we just paid a $214 phone bill, or Tom Harris made a $240 sale to the Jones Lumber company. Data such as this goes into the computer. The computer correlates this data, rearranges it and combines it into a useful form. The output of this would be information. Information for our purposes here is simply correlated data, classified data, or summarized data.

Figure 2 shows how a computer fits into an overall business operation. The manager or owner of a business, of course, is at the top of an operation, and the diagram shows data going into the computer from the manager and also from the organization itself, the organization being composed of individual people. The diagram also shows information going back to the manager, to the organization or the people in the organization. The computer in no way eliminates the communication that takes place between the manager and the organization.

If you look at a typical business operation, you will find they are inundated with paper work, phone calls, etc. If the paper work and phone calls reach a sufficient volume, the business can get to a point where one can't see the forest for the trees, and instead of proceeding in an orderly fashion, the manager and the organization itself can be operating in a mode where it just handles the first emergency that comes up.

A computer itself is not going to put order into any such confusion in a business. When a business decides to get a computer, the preparation that is involved in order to install the computer forces discipline on the business itself. The computer has to have things presented to it in a very precise manner in order to operate at all. The mere act of getting ready to install a computer system can put enough order into the business and increased its efficiency to such a point that it more than pays for the cost of the computer system.

Take another look at Figure 2. Figure 2 shows that the computer makes information available for the manager of the organization and also to the employees of the organization. This increased availability of information should make the organization more productive, should increase the availability of its products and should smooth out the interaction of the organization with its environment — its customers, its vendors, and the people in the neighboring community.

Figure 3 shows the menu for the Business Management System that we use on the AM-100 computer. A menu is simply a list of options. We will go through these options and briefly describe what each one does.

Overview of A Business Computer System

By James W. Kitzmiller

Figure 1. Fundamental action of a business computer system.

Figure 2. Use of a computer in a business.

```
WHICH WOULD YOU LIKE TO DO?

0  END
1  ORDER ENTRY
2  INVENTORY CONTROL
3  SALES ANALYSIS
4  ACCOUNTS RECEIVABLE
5  ACCOUNTS PAYABLE
6  GENERAL LEDGER
?
```

Figure 3. Menu for the AM-100 Business Management System.

ORDER ENTRY

Option number one is order entry. Order entry is the process of telling the computer that a customer has bought a certain item or set of items from your business. The data that you would feed into the computer for an order entry would be the same data that your salesman would put onto a sales slip when he making a sale; this data would be who the customer is, and his address. You would also include the name of the salesman as well as what item is purchased, how many of that item, and what the sales price is as well as the total price. Please take note that this order entry provides data to the computer so that the computer in a separate action can provide information to the organization and to the manager.

INVENTORY CONTROL

Inventory is the collection of products that a business has for sale to its customers. Inventory control is the process of keeping track of how many of those items you have, when it's time to get more, where they are stored, how fast they are selling and so forth. The inventory control section of our AM-100 Business Management System performs a number of different functions. One is the actual process of entering data into your inventory data files on the computer. This is the act of providing data to the computer. Data that you enter would

be a part number for each item, a description of each item, name of the vendor who provides that item to your business, the purchase price for that item, the sales price for that item, how fast you sell that particular item, how many of those items you have in stock, how long it takes you to get that item from the time you place an order for the item until you receive it, and so forth. The inventory record also shows how many of those items you have in stock and also how many of those items you have ordered but have not yet received. Also, the inventory record contains data about how many of those items are back-ordered. Back-ordered means that you have made a sale to a customer but were unable to deliver the item right away because you did not have it in stock. The inventory control system would have the ability to allow you to add all of this data into the files as well as change any of this data.

The AM-100 Inventory Control System produces many reports. The first such report is an Inventory Status Report. That simply lists all of the data that was described above for each individual part in stock.

The Inventory On Order Report gives the business manager, salesman or purchasing agent information about which items have been purchased by the company but have not yet been received.

The Inventory Shortage Report gives a list of the inventory items that it is time to purchase. This is different from being out of an item. The computer program takes into account the fact that the items are being used at a certain rate and the fact that there is a certain time period from the time that an order is placed until the item is actually received. The Inventory Shortage Report gives a list of the items which should be purchased at this time from the vendors.

Another option of our AM-100 Inventory Control System allows the purchasing agent for the business to select which of the items to order at this time. He will use knowledge of the availability of cash and credit in order to determine which of those items should be purchased.

Another section of our Inventory Control System is the Physical Inventory Checklist. This is a list of inventory items printed on an 8½" x 11" sheet of paper that allows a stock clerk in the organization to go around and take a physical count of how many of each item are in stock. The result of this physical inventory will be used to make any corrections of errors in the inventory count that is stored on the computer itself. For example, if the computer says you have five items in stock but the physical inventory count shows there are only four, the business owner can correct the data that is stored on the computer.

Another inventory report is the Inventory by Value Report. This report shows the dollar value of each inventory item in stock. For example, your merchandise was worth a dollar each, and you have four of them in stock. It would show the item, four dollars worth of value; it also lists how many and the cost of each one. The computer sorts this data in to the order of total value for each type of item, so that merchandise of greater value would appear before that of lesser value.

Another feature of our inventory control system allows the user to print purchase orders. Rather than printing these purchase orders on a special type of form, we simply print them on a blank 8½" x 11" paper so that the user does not have to buy any special forms.

SALES ANALYSIS

A sale is the transfer of ownership of a company product to a customer and receiving monetary exchange. Analysis is the act of breaking something down into its parts.

The Sales Analysis part of our Business Management System performs three different types of Sales Analysis.

The first type is Sales Analysis by Salesman. This lists the dollar value of the sales for each salesman. Information such as this can be used by the sales manager to see which salesmen are doing the best, and he can find out which actions these salesmen are taking that make them successful.

Another report of the Sales Analysis System is Sales Analysis by Product. The business owner, business manager, or sales manager can tell which products are moving the best, and he can emphasize sales of that particular product and increase production in that area; he can also tell if sales of a particular product have dropped off.

The next section of the sales analysis system is the Sales Analysis by Customer. This allows the sales manager to determine which customers are buying the most. The sales manager can then create a sales plan to approach that kind of customer with additional products in order to increase sales.

ACCOUNTS PAYABLE

Accounts payable is just a list of the people to whom you owe money, and it tells you how much you owe each person.

One feature of our Accounts Payable System is vendor file maintenance. Maintenance is the act of keeping up to date. A file is a collection of data records like a card file. A vendor is a person from whom you buy products. You can maintain the data about vendors through this section of the Business Management System. A record is a set of data about a particular item. You might keep a record on a 3x5 card. The system allows you to add records to the vendor file, delete records, change records, list vendor records on your terminal or list vendor records on the system printer.

The major report of the accounts payable system is the Aged Payables Report. This is just a list of the bills that you owe by age with the oldest bills listed first on the report.

Another report is the Accounts Payable by Vendor Report. This just lists how much you owe to each vendor.

Another section of the Accounts Payable System allows you to select which bills to pay at this time. The dollar value of these bills is subtracted from the cash on hand, and the fact that you have paid those bills is entered into the Accounts Payable System.

ACCOUNTS RECEIVABLE

Accounts Receivable is a list of who owes money to your company. It operates very similarly to the Accounts Payable. The first feature of the Accounts Receivable system is the Customer File Maintenance. This allows you to add customer records, delete customer records from your list, change data in the customer file, print out the customer records on the user terminal and print customer records on the system printer.

Another feature of the Accounts Receivable System is invoicing. Whenever an order is placed by the Order Entry System, the data is prepared so that invoices can be made and sent to the customer. These invoices are printed on standard 8½" x 11" paper, and the customer's name and address is inserted on the invoice at a position where it will appear in a window envelope. Your company name and address will appear on the invoice; the computer will print this for you.

The AM-100 Business Management System Accounts Receivable Section also will print statements for you. Once a month or as often as you choose, the system will

accumulate records of all unpaid bills owed to you and send out statements to each individual customer.

GENERAL LEDGER

A ledger is a book where you keep records of various ways that your business took in money, spent money, and it also lists various things that your business owns and the various places where your business owes money. There is a separate page for each type of expense or each type of income and so forth. A general ledger is a book of this type.

The General Ledger System allows you to tell the computer where you received money or where you spent money and how much; it also allows you to tell the computer how much dollar value there is on each thing that you own, such as your office furniture or your office building. You can also tell the computer how much money you owe on various things, such as the mortgage on your building. All this is the data input to the General Ledger.

The General Ledger System consists of numerous reports. One such report is simply a concise printout of the data that you entered into the computer.

Another report is a listing of each of the various areas or categories that you used to define the assets of your business, your income, expenses and so forth, and it lists how many dollars you have allocated in total to each of those areas. Another report is the Profit and Loss statement sometimes referred to as an Income Statement. This just shows you how much money you have made over a specified time period, such as the month or the year.

> ## This information enables the business owner and the employees to act in a co-ordinated fashion. . . and do co-ordinated planning. . .

The General Ledger makes a report called the Capital Statement. Capital is the net worth of the owner of the business or the net worth that the stockholders have in the corporation. A Capital Statement shows the increase in net worth over a given time period.

Another major report of the system is called the Balance Sheet. The balance sheet shows all of the assets of the business according to category; it lists all of the liabilities or money owed by the business; and it lists the capital of the business all according to the fundamental accounting equation: assets equals liabilities plus capital. So the balance sheet details each of those categories where the assets are, where the liabilities are and where the capital is.

CONCLUSION

This is just a very brief overview of what a computer system such as AM-100 Business Management System does. Basically, the business owner and his employees feed data into the computer and get out information. This information enables the business owner and the employees to act in a co-ordinated fashion.

No computer system is going to do the thinking for the business manager or his employees. The computer will give information to the manager, and the manager with his judgement and perception of the environment will use this information to make the correct decisions. □

GLOSSARY

Account	A category (type) of income, expense, asset, liability, capital, etc.; a record of activity in such a category.
Accounting	The process of recording, categorizing and summarizing data into a useful form. Although it is usually concerned with finance, the same concepts apply to other areas.
Accounts Payable	A list of the people who are after your money.
Accounts Receivable	A list of the people who owe you money.
Analysis	The act of breaking an area into smaller parts and learning more about the area by studying the parts.
Computer	A device that reads in data, stores it, rearranges, makes computations and tells you the results.
CRT	The TV type screen with the keyboard. (Abbreviation for cathode ray tube — an electronics term.)
Data	Records of details of events that have taken place.
Environment	Surroundings including customers, prospects, competitors, the neighborhood, and the prevailing laws.
General Ledger	A ledger used for accounts of a general nature. There can be "non-general" ledgers such as an accounts payable ledger.
Information	Data that has been aligned, categorized and/or summarized.
Inventory	Items in stock that are to be sold.
Inventory Control	The process of keeping track of the purchases, storage and sales of inventory and using that information to optimize purchasing schedules and quantities.
Journal	A place where you record transactions. In manual accounting systems, a journal is a book.
Ledger	A recording of accounts with records of each account kept in a separate area. Usually a ledger is in the form of a book. Each page contains information on a particular account.
Menu	A list of choices such as in a restaurant. The menu appears on the computer screen.
Order	A request from a customer to purchase your products.
Order Entry	The act of informing the computer that you just made a sale. It includes details of the sale.
Product	That which is brought into existence by your company and can be exchanged with the public for money or other means of survival.
Sale	The act of causing someone to buy your products and receiving a monetary exchange.
Sales Analysis	The process of studying sales when categorized into areas such as territory, type of product, type of customer, or salesman.
System	A set of machines, people and/or policies arranged to produce a desired product.
Transaction	An interchange such as a sale or a purchase.

The Automated Attorney

By Mathew Tekulsky

Tom Lambert's Century City, California law office, which specializes in aircraft accident suits, has no full-time secretary. Instead, there's a microcomputer sitting in the corner. With this computer Lambert feels his office is "on the leading edge of the frontier in this particular use of office equipment. Computers are getting into a price range that relatively small law offices can afford, and the capabilities of the machines, if used efficiently, make the law practice of a much higher quality."

This means a lot more than just typing letters, as the computer has captured the central role in Lambert's three-man law office. One function of the microcomputer is the analysis of pertinent data in pending cases. For example, there is a family of three or four programs that are designed to calculate "the dynamic rollover" phenomenon of helicopters and to make a general quantitative evaluation of a sudden loss of tail rotor thrust.

"The program is basically an inquiry into the ground handling stability of any helicopter on any particular type of terrain," Lambert explains. "When we put in the specific data for a particular helicopter, it told us in effect that we had neither a dynamic rollover nor a ground pitching instability condition operative to cause the helicopter accident that we were working on. We use it as an exploratory tool and itRs somewhat unique to our office."

But then, it's office is somewhat unique. All three lawyers are professional pilots, and two out of three are mechanical engineers by trade. Consequently, they do the bulk of their research in-house. Once they understand a problem, they go to outside experts for confirmation, review and preparation of expert testimony for the trial.

Another family of programs which Lambert developed involves building a mathematical model of applied loads for the "tail feathers" of a helicopter which had been involved in an accident, and then relating that to the actual loads of the components that broke.

Lambert explains how it works: "We calculated each of the failure modes of the component, and then another program compared the applied load to the resulting failure mode. As soon as a resulting failure mode occurred, it would plot a data point. This gave us a whole family of curves that told us which part of the system would fail first and what type of a failure it would be. Then we took the strength of the component and deteriorated it from 100% all the way down to 33%."

The 33% is significant because that is the FAA required margin of safety. For tactical reasons, Lambert did not use this information as hard evidence during a recent trial. However, it was very useful in the preparation of litigation.

His third family of programs is a damage evaluation analysis. "I've never really been totally satisfied with the accuracy of how actuaries prepare projected economic loss for wrongful death cases," Lambert says. "This family of programs is designed to assess the economic loss of the plaintiffs in wrongful death and personal injury lawsuits. It will also assess the value of the 'loss of companionship' in a wrongful death case and the 'pain and suffering' concept in a personal injury case."

The program itself contains certain variables such as the victim's vital statistics: age, the probability of survival to a particular age, and dates of birth and death. Then the computer estimates the individual's future rate of earnings and rate of return on invested capital, takes a per capita basis of reduction for personal consumption, and whatever is left over belongs to the survivors.

The program calculates the loss of companionship in a wrongful death situation on a per diem basis, i.e. how much per day. It does the same thing with the pain and suffering concept in a personal injury case which, according to Lambert, can always be related rationally to some value of dollars over a projected period of time, usually starting high nearer to the injury and stabilizing over the individual's projected life span.

"The big advantage of using the computer for this type of evaluation is that you can do quantum jumps," says Lambert. "With this method, you look at each year in the individual's projected life span individually as opposed to taking an average over a life span. The result you're trying to find is the basic economic loss that is solidly, economically and factually justified."

Due to the diversified nature of aircraft accidents, which often involve international parties that must be treated on an individual basis, this program is extremely useful to Lambert.

"The variables that go into this are enormous, so we take the generalized case and apply it to specific situations," he says. "And something in a generalized format like this is particularly important when you get into situations in which you have a wide range of rate of return of invested capital. For instance, with people who are earning and investing their money outside of the United States, we have to look at the conditions that are applicable to them in order to determine how their families have been deprived by the loss.

"The difficult conceptual task here is to be able to foresee what the variables are in a generalized case for your programming, and then write a program that doesn't use up all your memory by taking into consideration variables that aren't needed. You want it to be as concise and crisp as possible, and you want to have it in a simplified format so you can use it and explain it to people who don't understand these concepts generally anyway."

This is the jury in many cases, although the information is useful from both the plaintiff's and the defense's point of view. For his output, Lambert employs a tabular format and a curve plot, the latter of which offers a graphic view of the validity of the data.

How does this work in court?

"Once you have the family of curves, you can have your economist come on and testify that the rate of interest in the future will most likely be a certain percent, allowing a little leeway on the low side and on the high side," he explains. "Then he could evaluate with some factual backup what the projected rate of application of earnings would be. At the point where these two parameters cross on the curve plot, you have defined a range of hard data as to what the projected economic loss to that individual is on the most rational and logical basis."

Another program that Lambert has just started writing is an attorney time-keeping system with features that permit adaptation to existing accounting systems and the immediate retention and recall of all information that's within a client's file. This would include how much time has been spent on certain cases, the aging of a client's account, and information pertaining to the preparation of periodic client billings.

As a result, billing will be easier and will show a continuous account of what has transpired in each case. This is cumbersome to do by hand and would otherwise have to be turned over to a bookkeeper.

In the area of word processing, Lambert has found many applications for his computer. "One of its very great features is the outlining capability," he says. "It's really good to sit down and organize your thoughts in outline form, hit the research, bring the results back and dictate the body of the brief for the secretary to type up."

The computer's text editing capability allows "Points and Authorities" (the legal terminology for "mini briefs" submitted to the court) to be stored on disks, edited and made applicable to the problem at hand. Lambert puts everything onto a disk initially, and after it's reviewed and edited, he decides whether it's worth saving. The computer can also save case captions (names of plaintiffs vs. names of defendants) which the secretary only has to type out once.

Lambert explains how all of this facilitates the writing of a legal brief: "With each new pleading, we just load in the caption and the rest of the pleading from some other file in pieces and in a certain order from the disk into computer memory. Then when we're all through, we can add something special or unique to it, edit it, and when we've got everything we want, we just put in a signature line and a date. That completes the document and we can save that in a case file by putting it into the disk under its own file name. Then we print out the hard copy, reproduce it and serve it. It's a much quicker way to perform the very tedious tasks that are normally required in this type of practice.

"The big advantage here is that if you have some small error in the middle of the text, you don't have to rewrite 15 pages. You just go back in, clean up the mistake, get back out of the edit mode and run out a new hard copy. It only takes about 10 minutes."

THE SYSTEM

Lambert's system consists of a COMPAL 80 computer with 32K memory, a Multiterm printer, a 17" Sanyo television screen, a single disk Micropolis and a cassette tape recorder. The total cost is about $8,300, which includes some supplies like print wheels and ribbons as well as two software programs: Micropolis BASIC and the WORDPAL word processor.

The Multiterm printer, he says, is comparable in price and quality to the Xerox Diablo, except that the former comes equipped with a forms tractor that would ordinarily cost an extra $300. It also has a better graphics capability, but otherwise, "there's not a great deal of difference."

The reason he has both a cassette and a disk is twofold: first, he uses the cassette for a backup and second, much canned software available today comes in cassette form, particularly in the numerical area.

As far as software is concerned, Lambert has written over a dozen programs. In addition, he uses the WORDPAL to build text files which are saved and recalled in a variety of combinations. Owing to the somewhat unique nature of his practice, the computer's ability to handle both numerical analyses and text editing is its strongest attribute.

Lambert did not take any programming courses. He just studied the manuals, sat down and started writing. "I used to use basic programming when I was in the aerospace field," he states, "but there is still somewhat of a learning curve.

"If one is going to do the programming oneself, there's only one way to do it — roll up your sleeves, spend some time with the manual, spend some time writing programs and *do it*. Until you learn, you're going to use a lot of time that you may not have, and once you

get there, it's like flying an airplane — you have to stay current and proficient by updating, expanding and writing a new program every month or so."

The alternative to this, he says, is to go to a local university or computer club or go to your local computer store, hire a computer consultant, work out what the objective of the program is and have the consultant sit down and write it. This may or may not cost a great deal, depending upon the scope of the program objectives.

In terms of quantifying the value of his own programs, Lambert "wouldn't even know where to start."

"It's such a unique application, there may be no marketplace for it," he says, "and yet when you find a marketplace, your opposition may be willing to pay ten times the legitimate price for it to eliminate the surprise element."

Lambert has had his computer for about four months and although he is not taking advantage of every feature the machine has to offer, the computer was an improvement over his existing method within a week after the system was purchased.

"The WORDPAL part of the system is far and away the easiest to learn, and it's the quickest," he says. "The text editing can be learned by any competent secretary in no more than a week's time."

THE REAL WORLD

Lambert offers some advice on how to obtain maximum efficiency with one's computer. "With the WORDPAL, the first order of business is to work out in advance a system of allocating your disks," he says. "It's very easy to save everything and fill up your disks, but if you have to go back and search each of the directories on 20 disks, you haven't really accomplished anything.

The disks are sufficiently inexpensive that if you're going to err, err on the side of having a few empty spots to fill up. We have assigned one disk for internal office use, another for numerical programs, which I'm expanding rapidly now into one disk for every major numerical family of programs, one for each major case and one for miscellaneous cases."

The next step, he believes, is to establish in advance a system of naming files, which only have 10 characters. Lambert's method, adapted from law library citations, uses two numbers followed by a dash, then three letters followed by a dash, and finally three more numbers or letters. For example, 32-CAP-1 identifies the first "caption" of case number 32. Each case has its own case number for the file name identification.

The greatest asset of the computer, according to Lambert, is "saving time."

"It's hard to say just how much we have saved, but it certainly has been substantial," he says. "We're now using just one high quality (and expensive) regular part-time secretary and a couple of others on an overload, part-time basis."

In addition, the computer adds to the quality of Lambert's law practice with its ability to produce perfect copies, and it helps him better understand the phenomena at play in particular accidents.

"Since much of our work involves reconstructing and understanding the reconstruction of aircraft flying machine accidents, including airplanes, helicopters and even hang gliders," he says, "if it does nothing more than help me to better understand the evidence so I can present it better, then it's done its job."

Indeed, Lambert's research into the ground stability phenomenon in helicopter accidents and other situations may represent a significant contribution to the aerospace community which can be attributed directly to his computer. □

Hard Copy: Why Not the Best? Go Daisywheel!

By John MacDougall

If you are thinking of word processing either for letter writing, for legal manuscripts or just for getting publishable assembly listings, you are also thinking about some kind of quality printer. One of the highest quality printers on the market today is the daisywheel printer made by Diablo and Qume. This printer mechanism has the advantages of speed (30 cps), variety of print fonts, variable line and character spacing, and finally, a mechanism which is extremely simple and very compatible with electronic interfaces.

Recently, these mechanisms have begun to appear on the surplus market at prices which are attractive to the serious hobbyist. There is very little published information about these printers although there seems to be a strong interest in their use. Over the last year and a half I have built a succession of daisywheel interfaces for my own use. This article describes the latest of these. The interface described here is the simplest driver reasonably possible for a word processing application. It has been in use for several months now and is eminently suitable as a "starter" system.

DESCRIPTION OF THE PRINTER MECHANISM

The description which follows applies directly to the Diablo "HYTYPE I" daisywheel printer mechanism. This is the one I have and appears to be the only type on the surplus market. Recently, Diablo and Qume have both introduced microprocessor controlled printers of the same general mechanical characteristics. Interfacing with these is quite another story.

The HYTYPE I has a few moving parts. The first of these is the platen. This can be either a friction feed or sprocket feed device and can be controlled in ¼₈ of an inch increments either up or down. The platen is driven by a gear coupled stepping motor.

The second moving part in the HYTYPE I is the carriage. The carriage is driven by a large servo motor which pulls it equally well in either direction with a loop of cable. The servo is a sophisticated analog/digital hybrid, and the carriage can be zipped the full width of the platen in less than 400 milliseconds. One of the advantages of this system is apparent in "tabbing" operations where the carriage literally jumps from position to position without the slow jogging of the usual stepping system. In these conditions the HYTYPE I can be faster than its rated 30 characters per second.

The third moving part in the HYTYPE I is the printwheel. This is driven directly by another small servo motor and is the only one of the moving parts which has positional memory. The printwheel mounts directly on the servo motor shaft and can be easily and quickly interchanged upon tilting the carriage mechanism back.

There are two other moving parts in the mechanism both on the carriage. The first is the hammer which strikes

the print wheel "petal" to make the type impression, and the second is the small stepping motor which pulls the inked ribbon.

The servo systems which drive the carriage and printwheel in the HYTYPE I are very sophisticated electronically. In addition, the system is interlocked electronically so that, for instance, the carriage cannot move while the hammer is striking the printwheel. A number of other functions are carried in the electronics, as we will see later, but the net result of all this is about a square yard of circuit board spread along the bottom of the machine and up the back. The boards are stuffed, for the most part, with standard TTL integrated circuits, with a few operational amplifiers and FET switches thrown in. The data input lines are loaded with 250 ohm pullup resistors and protected by diodes to + 5 volts and ground.

> ## . . .the eleven bit machine is simpler, and modification of the hardware and software. . .for the eleven bit machine is an easy reduction from the illustrated material.

The servo drives for the carriage and printwheel derive their high slewing rate from high current, high voltage power supplies. In the new HYTYPE II's the power supply is a switcher and easily fits within the frame of the printer. In the older machines on the surplus market, the power supply is a large conventional brute which weighs about 40 pounds and is hard to hide. Fortunately, the power supply has also appeared on the market at a reasonable price, or you would be faced with constructing a unit with ± 15 volts at 9 amperes and 5 volts at 5 amperes capability. It's worth buying since it also has crowbar short circuit protection and is interlocked to take all of the supplies down if one fails or is shorted.

HYTYPE I mechanisms are available on the surplus market integrated with a keyboard at about twice the price of the parts alone. If you are not willing to go to some trouble in constructing interfaces and software drivers, you had better stop reading and get one of the complete units. Just to emphasize a point, the extreme versatility of the HYTYPE I has the penalty that the driver software must do everything. For instance, the carriage movement is incrementally controlled, and it is up to the controller to remember the carriage position so that the correct data can be applied for the carriage return. Otherwise the carriage will probably crash quite happily, and at high speed, into the end stops.

The basic unit, unlike mechanically driven terminals, is entirely separate from the keyboard and only associated with it through the interface electronics. You thus have the option of making a complete unit or of using the HYTYPE I as a printer only (as I now do). Optionally,

if you want to make a complete terminal, you can use any reasonable keyboard of your own choosing. Although I don't use it, the interface described in this article has the provision for a keyboard input.

The data connections for the printer are shown in Table I. The data input is carried on eleven or twelve lines depending on which machine you bought. By far, the most common data bus is eleven bits wide. My machine happens to be one of the rare twelve bit machines. For interface the eleven bit machine is simpler, and modification of the hardware and software in this article for the eleven bit machine is an easy reduction from the illustrated material. The table shows my understanding of the meanings of the various data lines for the two types of data busses. If this doesn't seem to fit what you have, then you will have to contact Diablo with specific details of circuit board number and revision for further information.

Table I. Data Connections for HYTYPE I™ Printers

Connector Pin Designation	Signal (All signals active low)	Machine Character Print Data	
		11 Bit Bus	12 Bit Bus
a. Input to Printer:			
h	DATA 1	0	0
j	DATA 2	1	1
m	DATA 4	2	2
f	DATA 8	3	3 Character Code
k	DATA 16	4	4
i	DATA 32	5	5
g	DATA 64	6	6
d	DATA 128	X	R
b	DATA 256	X	R Ribbon Adv. Code
V	DATA 512	X	R
F	DATA 1024	X	H Hammer Pres. Code
L	DATA 2048	not used	H
C	Platen Strobe		
K	Carriage Strobe		
P	Print Strobe		
M	Ribbon Lift		
E	Restore		
S	Select Printer		
H	Select Ready Lines		
b. Output from Printer:			
a	Printer Ready		
B	Check		
R	Paper Out (only works if switch is added)		
c	Platen Ready		
w	Carriage Ready		
Y	Print Ready		
Z,n	Not to be used		
Note: "X" means "don't care"			

Data for the carriage movement and the platen are applied in a similar manner. The highest order bit determines the direction of movement, and the lower bits determine the amount of travel. Notice that this is not a two's complement or other similar mathematical number. It is very simply a positive number giving the distance of travel and a bit which determines the direction of travel. A "one" in the high order bit means carriage movement from right to left or a platen movement in the reverse direction to a normal feed line. The distance of travel indicated by the lower bits is a multiple of fractions of an inch.

In the twelve bit machines the fundamental increment is $1/120$ of an inch horizontally and $1/48$ of an inch vertically (platen). In the eleven bit machines the fundamental increment are $1/60$ of an inch for the carriage and $1/48$ of an inch for the platen. All of the illustrations and examples in this article are for the twelve bit machines. For instance, if you wish to move the carriage the correct increment for 12 characters per inch, the calculation is done as follows:

$$\text{Number of fundamental increments} = \frac{\text{Number of basic increments per inch}}{\text{Desired characters per inch}}$$

Thus, for 12 characters per inch in a twelve bit system

Data = 120/12 = 10 decimal = A hexadecimal =

= 0000 0000 1010 binary

This is exactly the data which would be put on the data bus. If the movement were for a backspace, the reverse bit would be set, and the data would appear as 1000 0000 1010. Similar reasoning applies for the platen movement.

The character data is different. In this case the code for the actual character is put on the data bus, and the printer has the necessary internal electronics to determine the print wheel movement needed to find the correct character. In the twelve bit machines the upper data bits also determine the amount of hammer pressure and the amount of ribbon advance. I have incorporated this in my software. The upper bits of the character data have no meaning in the eleven bit machines and can be ignored.

Notice that in all of this discussion, the same data bus is used no matter what the function. The function that the data performs is determined by the strobes. There are three strobe lines, and only one strobe line can be strobed at a time. However, the machine will remember, and it is possible to strobe a carriage movement, for instance, while a print is taking place.

Before a function can be strobed, however, the associated 'ready' line must be checked. There are four ready lines; three for each of the main functions (platen, carriage, printwheel), and the fourth shows if any of the other three are not ready. As soon as a ready line is down, the function can be strobed even if another function is already taking place. The strobe must be at least two microseconds long which is ideal for a micro system since that is about the shortest pulse it is possible to get out of an output port. The data must be present on the data lines before the strobe is applied and held until after the strobe is removed. There are three strobes, one for each function.

In addition to the data lines, the strobe lines and the ready lines, there are lines for selecting the printer and selecting the ready lines. There is also a line for paper out condition (this requires some kind of paper out switch to be effective) and a check line. The check line is used as a fault indicator to show that some false condition has been activated. The only signal which will be accepted by the printer after a check signal is the 'restore' command.

The restore command is used to reinitiate the printer mechanism after some kind of fault. The most common fault the beginner will encounter is a crash of the carriage into the end stops. I have brought the fault line out to a pushbutton on the front of the printer. When power is first applied to the printer, the electronics automatically initiate a restore sequence. A restore sequence moves the carriage to the extreme left edge and then back about ½ inch to the right of the left end stop and then sets the printwheel in an initialized condition so that the electronics know where the print wheel is. Sensing of the printwheel 'park' position is done by a special magnetic pickup and a small metal slug on the position transducer.

There are two position transducers: one on the back of the carriage servo motor and one on the back of the

printwheel servo motor. These are closely coupled high frequency sensors. Adjustment of the position transducers calls for specialized knowledge and equipment.

One final input is the 'ribbon lift'. In the original version of the HYTYPE I this was intended for use with two-color ribbons. However, the way the ribbon is mounted in front of the printwheel, it is impossible to see the most recently typed data. In my system I use the ribbon lift command to position the ribbon in front of the type wheel just before a print is to occur. After a spell of typing, there is a pause before the ribbon drops down to reveal the text. This pause, which is controlled by a retriggerable monostable, prevents the ribbon from continually popping up and down in a most annoying manner. There is nothing unique about this scheme, and it is used in several other common terminals and in some of the recent HYTYPE II terminals. Of course, for this to be effective you will only be able to use one-color ribbons.

To summarize the data requirements of the HYTYPE I: there are 11 or 12 data lines, three strobes and a ribbon lift going to the printer from the computer, and four ready lines going from the printer to the computer. Thus, dedicated to the needs of the printer are two full output ports and one half of an input port. The other lines are not necessary for computer control and therefore don't require data ports. If your system has parallel ports already available, you don't need a special interface card and can skip the next section for the time being.

A photograph of the printer is shown in Photo 1. Note the home-built paper roll holder. The ribbon comes in a cassette form, and different colors as well as carbon ribbons are available. A spare daisywheel can be seen in front of the printer beside the interface board.

PHOTO 1 Photograph of the HYTYPE I showing the Interface Board and a spare Daisywheel in front. Note the additin fo the rack for holding rolls of paper. The flat cable for interconnection to the driver electronics is shown at the right.

HARDWARE INTERFACE

As I said before, if you have a couple of good drive capability output ports available, you can skip the hardware discussion altogether. I chose to build an interface card for a number of reasons, the chief one of which was that I didn't have any parallel ports to spare. Besides, with the use of a wire wrap tool, some sockets and a Vector S-100 board, it's kind of relaxing and a change from the daily drag to immerse oneself in mundane wiring problems. Another reason for using the interface board was that I wanted to incorporate a small separate dedicated RAM area for later use as buffer storage. As it turned out, I don't use this buffer as such but it has sure come in handy for storing small programs such as memory test routines when I want to preserve my main memory completely free.

A final reason for using the dedicated interface was that I wanted to use a code conversion PROM between the data input and the printer. This allowed me complete freedom in the use of keyboard and input data and the printwheel characters.

A summary of the input data required to drive the printer was given in Table I along with a summary of the printer output data. All of the printer input data lines have 250 ohm pullup resistors, so you must provide driver devices with good current capability. Open collector or tri-state devices are equally effective.

The complete circuit diagrams for the interface cards are shown in Figures 1 and 2. The card is functionally divided into five areas. These are:

1. Address and Port decoding logic
2. RAM and associated buffers
3. Printer driver hardware and computer output ports
4. Computer input ports and data multiplexing circuitry
5. The ribbon lift circuit

The port selection logic and the RAM address decoding logic share some common parts, and both may be relocated together by means of the DIP switch to any 4K boundary. Location addresses and switch positions are given in Table II. In my system the RAM resides at A000, and consequently, the printer output ports A0 and A1. Note the port assignments on the circuit diagram.

Since my system is getting pretty full, I am now putting buffers on all S-100 bus card inputs. The address input buffers are 74LS367 operated in the non-tristate mode. The buffers on address inputs A9 and A8 are also used to drive the lower data bits of the port address decoding. The RAM devices are 2102, and chip select is

Table II. Eight Position DIP Switch Coding

a. Address Selection: ("0" = "on")

Switch Number				RAM Starting Address	I/O Port Numbers
4	3	2	1		
0	0	0	0	0000	00,01
0	0	0	1	1000	10,11
0	0	1	0	2000	20,21
0	0	1	1	3000	30,31
0	1	0	0	4000	40,41
0	1	0	1	5000	50,51
0	1	1	0	6000	60,61
0	1	1	1	7000	70,71
1	0	0	0	8000	80,81
1	0	0	1	9000	90,91
1	0	1	0	A000	A0,A1
1	0	1	1	B000	B0,B1
1	1	0	0	C000	C0,C1
1	1	0	1	D000	D0,D1
1	1	1	0	E000	E0,E1
1	1	1	1	F000	F0,F1

b. Hammer Intensity Selection:

Switch Number		Relative Pressure
6	5	Intensity
0	0	1
0	1	2
1	0	3
1	1	4

c. Spacing Selection:

Switch Number		Character Spacing	Line Spacing
8	7	Characters/Inch	Lines/Inch
0	0	10	3
0	1	10	6
1	0	12	3
1	1	12	6

Figure 1. Schematic of part of the Interface Card showing the Address Decoders and Buffers.

Figure 2. Schematic of the remainder of the Interface Card showing the I/O ports, the Data Buffers and the MUX Circuitry.

derived from the high order address bits and the processor SINP and SOUT lines. The RAM output data is multiplexed into the computer input data bus, as will be described later.

The necessary 16 bits of output required to drive the printer is generated through two 8212 output port devices. These have sufficient capability to drive the 250 ohm pullup resistors if necessary. In my system I have a code conversion PROM in the lower eight bits of the 12 bit data bus. This PROM is enabled by the ribbon lift signal and held on during the printing of a character. For all other outputs to the data bus, the PROM is bypassed by the 74LS367 tri-state buffers, and the lower eight bits appear on the data bus exactly as sent. If you don't want to do any code conversion, this bit of circuitry can be eliminated, or you can do any necessary conversion in the software. Incidentally, this conversion scheme is very handy for converting what-have-you to ASCII and vice versa. The 93448 PROM has room for two complete sets of code.

The computer input must be serviced from three sources on the card. These are the RAM output, the keyboard output and the status byte.

The keyboard output is combined with the RAM output on the same data lines by means of their respective tri-state outputs. This common data bus is one set of inputs to the 74LS157 multiplexers. The other set of inputs comes from two sources. The upper four bits come from the remaining four switches on the 8-bit address decoding DIP switch. These four switches are used to set the character spacing, the line spacing and the character impression intensity, as will be described later. The lower four bits are provided by the four "ready" lines from the printer. These can thus be interrogated by the software. Selection of the computer data sources to be put on the S-100 bus is determined by the address decoding logic.

The final block on the circuit card is the ribbon lift circuit. This circuit uses a retriggerable one shot multivibrator, the 9601, to provide a pulse which holds the ribbon up between the print wheel and the platen for a fixed period of time. Each time a character is typed, the ribbon hold period is reinitiated so that during rapid typing of extended passages, the ribbon never drops. The emitter follower is used for time extension with the RC combination shown. If you think the ribbon lift time is unsatisfactory, this can be changed by changing the value of C. I experimented with several times and found those shown to be best for me.

All the circuitry shown in the figure is contained easily on one Vector S-100 circuit card. I used two regulators, one for the RAM and the PROM and the other for the remainder of the circuitry. Since all of the IC's shown are five volt devices, only two 7805 regulators are required.

A photograph of the completed board is given in Photo 2. Note the space at the top of the board for an additional 1K of RAM and the space in the lower left corner for additional keyboard or I/O circuitry. A socket is also provided for a keyboard PROM, if required.

THE SOFTWARE

Now that you have the mechanical aspects of driving the HYTYPE I well in hand, the next step is the software. A complete listing of the software which I now use is given at the end of the article. This listing is for the simplest driver reasonably possible for the printer. If you want to do fancy things like changing the spacing dynamically, a more complicated program is necessary. I found that for the things which I do at the moment, the extra complications of the expanded software were not warranted.

PHOTO 2 Photograph of the complete Interface Board. Two ribbon connectors are used to connect the board to an adapter plug which fits the ribbon cable from the printer. The keyboard would also enter here, if used.

The flow chart which describes the philosophy of the program is given in Figure 3. On entering the driver with something to be done, the first thing the program does is save all of the system registers, initialize the strobes and finally look at the setup switches to determine what character and line spacings are to be used. The switch status is saved in register B for later use. All input to the printer driver is assumed to be ASCII, so the parity bit is stripped before proceeding.

The program is arranged so that the most common functions are tried first. Obviously the largest usage of the printer is for typing characters, so the program first looks for a printable ASCII character. How a character is printed is described later on. If a printable character is not found, the program next looks to see if the input was one of the functions: SPACE, BACKSPACE, CARRIAGE RETURN, CRLF, LINEFEED, REVERSE LINEFEED, or SET LEFT MARGIN. If none of these are found, the default is a simple return without any action. Before returning from the driver, the last thing to be done is to recover all of the registers which were saved on entry. For this simple program, the functions are embedded in the program rather than contained in a lookup table. Thus, the program jumps from routine to routine looking for a match.

As an illustration of how the printer is strobed, we will use the routine PRNAS which prints an ASCII character. This routine is flowcharted in Figure 4. In the 12 bit printer the hammer impression can be controlled by the software. In my implementation there is a limited capability to do this as controlled by the DIP switch whose position was stored in register B on entry. Bits B4 and B5 control the impression density, but these should be bits D11 and D12 of the printer data bus, and this would mean that they should be in bits B2 and B3 of the data going to port A1. This gets all sorted out by shifting the data twice to the right and then masking it with OCH to eliminate the extraneous material.

With the impression bits set, we have hardly started. The next action is to set the strobes for all the outputs to "off" before sending the data to port A1 (remember this port does the strobing as well as set the upper four data lines). Thus our busy data byte now gets FOH added in. The upper 3 bits set the strobes to "off" while the next bit is used to trigger the ribbon lift and also, in my system, to turn on the PROM tri-state outputs (and turn off the PROM bypass drivers). The data byte to port A1 now looks like 1111 SS00 where 'S' indicates switch impression control data. Now the data (ASCII) byte can be put on the data lines by sending it through port A0. All of the setup is now complete, and the actual print can finally be performed.

The strobing of any function takes place in two steps. First, the function ready line is sampled to determine if

Figure 3. Flowchart for the overall Driver Software.

Figure 5. Flowchart for the Control Routine which moves the carriage one space.

Figure 6. Flowchart for CARMOV — the Routine

Figure 4. Flowchart for PRNAS —
the Routine for Printing
an ASCII Character.

for Moving the Carriage.

The good guys still wear white.

Buying a computer is serious business. All computer courses, seminars, lectures, etc, stress one important point; know your needs and know your suppliers, be sure you can rely on them. Khalsa Computer Systems is a retail computer dealership that stresses honesty and integrity, not only as a way to do business, but as a way of life. Wearing white is the reflection of that committment.

We select and stock what we believe to be the finest low cost computer systems for home and business. From the Apple II with colorgraphics at under $1,000 to the AM-100, the best data processing system available under $10,000. We also supply DEC equipment from $15,000 to $100,000 for the ultimate in mini computer power.

Our staff can help you evaluate and configure systems that will solve your immediate problems and still allow for future expansion. In our showroom we provide separate booths for each computer system, plus reference books, periodicals, a classroom, lounge and an office oriented demonstration area. Our software group consists of experienced systems analysts and programmers who know how to solve your data processing problems. Our full time service department and field service offers advanced and efficient service and parts to back up what we say and sell.

It's worth the drive to Pasadena to see us. You can rely on us to serve you now and in the future, we're committed to it, it's our way of life.

The people in white; we're the ones you can count on.

Open 12 to 8, Tuesday to Friday
12 to 6, Saturday
Closed Sunday and Monday

KHALSA ☬
COMPUTER SYSTEMS INCORPORATED

500 Lake Avenue, Pasadena, California 91101   (213) 684-3311

Khalsa Computer Systems, Inc. has also been known as the Byte Shop of Pasadena since we opened in 1976.

CIRCLE INQUIRY NO. 84

the function is busy. When the ready line goes low, the function can be strobed. Referring to the hardware drawings, we see that the ready line for a character print is bit 0 from port A0. This is checked in a loop until it goes low, and then the strobe data byte is recovered from register D where it had been previously saved. The strobe is set by 'ANDing' with DE(HEX) so that the data now looks like 1101 SS00. This is sent to port A1 to do the actual strobe. Of course, the strobe pulse must now be removed, and this is done by replacing the missing bit by ORing with 20(HEX) and sending the byte to port A1 once again. Finally, the ribbon lift pulse is removed and the program jumps to the next step in which the carriage is advanced one space.

The routine which moves the carriage one space is called SPC1, (see Figure 5), and it contains two other routines which really do all of the work; SPCVAL determines the correct increment to move the carriage based on the switch input data, and CARMOV is a general program for moving the carriage an amount determined by the contents of register DE. As a final example of how the printer is controlled, we will now examine CARMOV.

Moving the carriage in the routine CARMOV involves the operation of putting the data on the lines and strobing, as is done with the print operation, and also keeping track of the absolute position of the carriage. Remember, in the mechanical description we noted that carriage movement is purely incremental and that extraneous means must be provided to keep track of the carriage position.

On enter CARMOV (Figure 6) the routine first stores the contents of register DE. This register has the increment data for the carriage movement. First the lower byte is complemented and sent directly to the printer via port A0, then the operations of masking and setting up the strobe bits are performed on the high byte. Here, the lower nibble contains carriage movement information (lower 3 bits in the case of the 11 bit machines). This is masked and the strobes-off information is added by ORing with E0. The byte is then sent to port A1 and temporarily saved in register C. The status byte is obtained from port A0 and examined in the loop until bit 1 becomes low. At this time the carriage strobe is inserted by ANDing with AF and the byte again sent to A1. The strobe is then removed as before.

Since the reverse direction is set by a bit in the highest data position and not by the type of number, it is necessary to check the highest data bit to see if the carriage movement data should be added or subtracted to the data contained in the absolute position register. After testing, the appropriate arithmetic function is performed and the new position data stored before the routine returns back to the main program.

There is one final piece of software which is needed for smooth operation of your printer. This is necessary because on turning on your system, it is possible for the strobe outputs to come up in any arrangement and for random data to be present in the carriage position storage location. In my system I have an initialization procedure which the system goes through when coming up from a cold start. I have added to this a small routine for setting up the printer parameters. A listing of this routine is shown in Program 1.

PROGRAM 1

```
          .RADIX 16
          .TITLE 'Monitor Initialization Routine'
          ;This routine is inserted in the Monitor
          ;initialization subroutine for the purpose
          ;of correctly initializing the data ports
          ;feeding the DIABLO HYTYPE I printer
          ;mechanism
0000' 3EE0  DINIT: MVI   A,0E0   ;DATA FOR SETTING
0002' D3A1         OUT   0A1     ;    STROBES TO OFF
0004' 21 0000      LXI   H,0000  ;ZERO THE CARRIAGE
0007' 22 A3FC      SHLD  0A3FC   ;    POSITION STORE
          .END
```

Sometimes, because of a blown program, it is necessary to restart the system. This would normally be done without turning the printer off, and since the restart routine zeros the carriage absolute position count, you could be left with a printer sitting half way across the page and the system thinking this was the left margin. The way out of this is to use the "restore" line. On my printer I bring this out to a push button on the front of the printer. This works well for me. However, you might like to have the restore button on the keyboard as I had in an earlier configuration.

Now you're on your own. I certainly hope that you will gain as much use from your printer as I have from mine. It's a super mechanism and deserves much more recognition from the hobbyist than it is presently getting. □

Table III. Special Printer "Control" Functions

Control Character	HEX Code	Function
h	08	Backspace
t	14	Reset the left margin
q	11	CR + LF
l	0C	Reverse linefeed

Note: These special functions are recognized by the printer driver software — not by the printer itself. It is the driver software which tells the printer how to perform these functions.

PROGRAM 2

```
            .TITLE  'DAISYWHEEL PRINTER CONTROL ROUTINE'
            .RADIX 16
            ;This program is written for Z-80 driven control
            ;of a Diablo HYTYPE I (TM) printer mechanism which
            ;has a 12 bit data input bus.
            ;The program uses a temporary storage RAM
            ;located between A3F0 and A3FF.  The program is
            ;also configured for use with an interface PROM for
            ;the print data. The data ports used are--
            ;Input  A0=setup sense switches and printer readies
            ;       A1=reserved for keyboard input (not used)
            ;Output A0=Printer and platen low data byte
            ;       A1=Printer and platen low data byte
            ;
0000' F5    DIABLO: PUSH   PSW    ;SAVE THE REGISTERS
0001' C5            PUSH   B      ;
0002' D5            PUSH   D      ;
0003' E5            PUSH   H      ;
0004' 3EE0          MVI    A,0E0  ;INITIALIZE THE STROBES
0006' D3A1          OUT    0A1    ;
0008' DBA0          IN     0A0    ;CHECK THE SETUP SWITCHES
000A' 47            MOV    B,A    ;AND SAVE  IN 'B'
            ;
            ;The input data byte is in register C.
            ;It is assumed ASCII
            ;
000B' 79            MOV    A,C    ;
000C' E67F          ANI    7F     ;STRIP PARITY
000E' 4F            MOV    C,A    ;
000F' FE21  CHAR:   CPI    21     ;PRINT BET 21H AND 7EH
0011' 3826          JRC    SPACE  ;
0013' FE7F          CPI    7F     ;
0015' 2822          JRZ    SPACE  ;
            ;
            ;The subroutine for printing an ASCII character
            ;
0017' 78    PRNAS:  MOV    A,B    ;GET THE SWITCH DATA
0018' 0F            RRC           ;SHIFT TO
0019' 0F            RRC           ;  CORRECT POSITION
001A' E60C          ANI    0C     ; MASK IT
001C' F6F0          ORI    0F0    ;RIBBON LIFT &PROM ENABLE
001E' D3A1          OUT    0A1    ;STROBES STILL OFF
0020' 57            MOV    D,A    ;HOLD THE PRESS. DATA
0021' 79            MOV    A,C    ;
0022' D3A0          OUT    0A0    ;ASCII BYTE TO PROM
0024' DBA0  PRN1:   IN     0A0    ;CHECK STATUS
0026' CB47          BIT    0,A    ; OF THE PRINT WHEEL
0028' 20FA          JRNZ   PRN1   ;
002A' 7A            MOV    A,D    ;STROBE THE
002B' E6DE          ANI    0DE    ;
002D' D3A1          OUT    0A1    ; PRINT WHEEL ON
002F' F620          ORI    20     ;    AND OFF
0031' D3A1          OUT    0A1    ;
0033' 3EE0          MVI    A,0E0  ;REMOVE THE RIBBON LIFT
0035' D3A1          OUT    0A1    ;
```

COLORADO
COMPUTER
CORRAL

November 11 & 12 1978

SAT 10 – 6

SUN 12 – 6

DENVER MERCHANDISE MART
451 E 58th STREET
DENVER, COLORADO

PRESENTED BY THE DENVER AMATEUR COMPUTER SOCIETY
FOR THE ADVANCEMENT OF PERSONAL COMPUTING IN THE
ROCKY MOUNTAIN REGION. WE WILL BE FEATURING A
ROUNDUP OF NATIONAL AND REGIONAL MANUFACTURERS,
DISTRIBUTORS AND RETAILERS OF PERSONAL COMPUTERS,
COMPONENTS AND SOFTWARE FOR YOUR VIEWING AND
PURCHASING PLEASURE. WRITE D.A.C.S., P. O. BOX 286
MORRISON, CO 80465   **ADMISSION $2.⁰⁰ at door**

```
0037' 1804            JMPR    SPC1    ;TO ADVANCE THE CARRIAGE
                      ;
                      ;These routines perform the various carriage
                      ;space  backspace and return functions.  CRLF is
                      ;a single command for both the CR and LF  thus
                      ;providing a potential program savings in external
                      ;programs and for local operations.
                      ;
0039' FE20    SPACE:  CPI     20      ;SPACE?
003B' 200C            JRNZ    BKSP    ;
003D' 1600    SPC1:   MVI     D,00    ;
003F' CD 009B' SPVAL: CALL    CARSP   ;GET THE PITCH
0042' 5F              MOV     E,A     ;
0043' CD 00B9'        CALL    CARMV   ;DO IT
0046' C3 0095'        JMP     RETN    ;DONE
0049' FE08    BKSP:   CPI     08      ;BACKSPACE?
004B' 2004            JRNZ    CR      ;
004D' 1608            MVI     D,08    ;SET REVERSE BIT
004F' 18EE            JMPR    SPVAL   ;
0051' FE0D    CR:     CPI     0D      ;CR?
0053' 2012            JRNZ    CRLF    ;
0055' CD 005B'        CALL    CARTN   ;DO A RETURN
0058' C3 0095'        JMP     RETN    ;
005B' 2A A3FC  CARTN: LHLD    0A3FC   ;GET THE CARR. POSITION
005E' 7C              MOV     A,H     ;
005F' F608            ORI     08      ;SET THE REVERSE BIT
0061' 57              MOV     D,A     ;
0062' 5D              MOV     E,L     ;
0063' CD 00B9'        CALL    CARMV   ;
0066' C9              RET             ;
0067' FE11    CRLF:   CPI     11      ;LINEFEED
0069' 2005            JRNZ    LF      ;
006B' CD 005B'        CALL    CARTN   ;DO THE RETURN
006E' 1804            JMPR    LINEF   ;
                      ;
                      ;The next routines perform the linefeed functions.
                      ;Note that the Diablo has a reverse LF and that
                      ;the platen does not need to move a line at a time.
                      ;
0070' FE0A    LF:     CPI     0A      ;LINEFEED?
0072' 2009            JRNZ    RLINF   ;
0074' CD 00A8' LINEF: CALL    LINSP   ;GET THE SPACING
0077' CD 00F2' LINF1: CALL    PLTMV   ;MOV THE PLATEN
007A' C3 0095'        JMP     RETN    ;
007D' FE0C    RLINF:  CPI     0C      ;REVERSE LINEFEED?
007F' 2009            JRNZ    RSETL   ;
0081' CD 00A8'        CALL    LINSP   ;
0084' 7A              MOV     A,D     ;
0085' F608            ORI     08      ;SET THE REVERSE BIT
0087' 57              MOV     D,A     ;
```

```
0088' 18ED            JMPR    LINF1   ;
                      ;
                      ;Utility to reset the left margin with control-t
                      ;
008A' FE14    RSETL:  CPI     14      ;RESET LEFT MARGIN?
008C' 2007            JRNZ    RETN    ;ERROR TRAP HOME
008E' AF              XRA     A       ;
008F' 57              MOV     D,A     ;
0090' 5F              MOV     E,A     ;
0091' ED53 A3FC       SDED    0A3FC   ;STORE CARR POSN
                      ;
                      ;Final return to program
                      ;
0095' E1      RETN:   POP     H       ;RESTORE STATUS
0096' D1              POP     D       ;
0097' C1              POP     B       ;
0098' F1              POP     PSW     ;
0099' 79              MOV     A,C     ;
009A' C9              RET             ;
                      ;
                      ;Routine to select the carriage spacing from the
                      ;DIP switch position.
                      ;
009B' 78      CARSP:  MOV     A,B     ;
009C' E680            ANI     80      ;
009E' FE80            CPI     80      ;
00A0' 2003            JRNZ    TENCI   ;
00A2' 3E0A            MVI     A,0A    ;12 CHARS PER INCH
00A4' C9              RET             ;
00A5' 3E0C    TENCI:  MVI     A,0C    ;10 CHARS PER INCH
00A7' C9              RET             ;
                      ;
                      ;Routine to select the platen spacing from the
                      ;DIP switch position.
                      ;
00A8' AF      LINSP:  XRA     A       ;
00A9' 57              MOV     D,A     ;
00AA' 78              MOV     A,B     ;
00AB' E640            ANI     40      ;
00AD' FE40            CPI     40      ;
00AF' 2004            JRNZ    LINS2   ;
00B1' 3E08            MVI     A,08    ;SET SINGLE SPACE
00B3' 5F              MOV     E,A     ;
00B4' C9              RET             ;
00B5' 3E10    LINS2:  MVI     A,10    ;SET DOUBLE SPACE
00B7' 5F              MOV     E,A     ;
00B8' C9              RET             ;
                      ;
                      ;Routine for setting the 12 bits of carriage
                      ;information on the data lines checking the ready
                      ;and strobing the carriage to move.
                      ;
```

```
00B9' ED53 A3F0  CARMV:  SDED   0A3F0  ;STORE THE POSITION STATUS
00BD' 7B                 MOV    A,E    ;
00BE' 2F                 CMA           ;INVERT THE DATA
00BF' D3A0               OUT    0A0    ;
00C1' 7A                 MOV    A,D    ;
00C2' 2F                 CMA           ;
00C3' E60F               ANI    0F     ;LOW NIBBLE ONLY
00C5' F6E0               ORI    0E0    ;STROBES OFF
00C7' D3A1               OUT    0A1    ;HIGH BYTE TO DIABLO
00C9' 4F                 MOV    C,A    ;SAVE IT
00CA' DBA0       CARDY:  IN     0A0    ;IS THE CARRIAGE READY?
00CC' CB4F               BIT    1,A    ;
00CE' 20FA               JRNZ   CARDY  ;
00D0' 79                 MOV    A,C    ;GET THE HIGH BYTE
00D1' E6AF               ANI    0AF    ;SET THE STROBE
00D3' D3A1               OUT    0A1    ;AND DO IT
00D5' F640               ORI    40     ;STROBE
00D7' D3A1               OUT    0A1    ;    OFF

                 ;This routine adds the carriage movement increment
                 ;(or subtracts it) to (from) the absolute carriage
                 ;address on completion of a carriage movement
                 ;
00D9' ED5B A3F0          LDED   0A3F0  ;GET THE MOVEMENT INCREMENT
00DD' CB5A               BIT    3,D    ;TEST FOR REVERSE
00DF' 2006               JRNZ   SUBTR  ;
00E1' 2A A3FC            LHLD   0A3FC  ;
00E4' 19                 DAD    D.     ;
00E5' 1807               JMPR   STORE  ;
00E7' E607       SUBTR:  ANI    07     ;MASK THE REV BIT
00E9' 2A A3FC            LHLD   0A3FC  ;
00EC' ED52               DSBC   D      ;
00EE' 22 A3FC    STORE:  SHLD   0A3FC  ;STORE THE CARR. ADDR.
00F1' C9                 RET           ;

                 ;Here the platen movement is set and strobed.
                 ;For this simple driver no record is kept
                 ;of the platen absolute address.
                 ;
00F2' 7B         PLTMV:  MOV    A,E    ;SET THE
00F3' 2F                 CMA           ;
00F4' D3A0               OUT    0A0    ;    LOW BYTE
00F6' 7A                 MOV    A,D    ;MASK AND COMPLEMENT
00F7' 2F                 CMA           ;    THE HIGH BYTE
00F8' E60F               ANI    0F     ;
00FA' F6E0               ORI    0E0    ;RESET THE STROBES
00FC' D3A1               OUT    0A1    ;    AND SEND THE HIGH BYTE
00FE' 4F                 MOV    C,A    ;SAVE DURING THE CHECK
00FF' DBA0       PLRDY:  IN     0A0    ;GET STATUS
0101' CB57               BIT    2,A    ;CHECK IT
0103' 20FA               JRNZ   PLRDY  ;
```

You can't beat our prices . . .
nor can you beat our delivery!

NORTH STAR
SAVE MORE THAN 15%

Complete Minifloppy Disk System
w/BASIC and drive, Kit List $699 . . . . . $ 589
   Assembled, List $799 689

4 MHz Z-80 CPU Board, Kit $199 . . . 169
   Assembled/Tested List $259 209

16K Dynamic RAM Board,
   Kit (order as 01-3216-0) List $399 329
   Assembled/Tested (01-4216-0) $459 379

CPM Disk Operating System Conversion
   Package for North Star, latest version . . . 145

HORIZONS	CASH PRICE
Horizon 1 Kit, List $1599	$1349
Assembled/Tested List $1899	1599
Horizon 2 (w/2 drives) Kit List $1999 . .	1699
Assembled/Tested List $2349	1939

SUPER SYSTEM

Horizon 1 Assembled/Tested w/2 serial
   ports, 1 parallel port, 6 extra edge
   connectors, List $2065 . . . . $1749

Horizon 1 Kit w/above extras List 1699 1429

Horizon 2 Assembled/Tested with
   above extras List $2515 2129

Horizon 2 Kit w/above extras (2099) 1779

SUPER BUYS ON MEMORY BOARDS

CROMEMCO — SAVE $300
16K RAM Board, factory assembled/
tested, List $795   OUR CASH PRICE $495

DYNABYTE 16K Static Board
   MSC1645, 450ns 329
   MSC1625, 250ns 359
   assembled, tested, guaranteed one year.

MORROW THINKER TOYS —
   Super RAM® 16K Static, Kit
   450ns List $299 . . . OUR CASH PRICE $269
For 250ns version, add $35. For factory assembly
and testing, add $35.

SPECIALS ON TERMINALS and PRINTERS

DYNABYTE 57 Naked Terminal
Assembled/Tested (order 03-4500-0)
List $350 . . . . OUR CASH PRICE $ 280

Intertube® 784

DECwriter II 1490
   Immediate delivey on all DEC printers

TELETYPE 43 Printers — new
. . . . AS LOW AS 880

MORROW Discus I®, assembled
   and Tested, List $995 849

Check our prices on Hazeltine 1500, 1400,
Soroc 120, and Lear Siegler ADM3A.

. . . . ONLY $ 769

Check our prices on Hazeltine 1500,
1400, Soroc 120, and ADM3A.

HAZELTINE 1400 Terminal (Assembled)
OR LEAR SIEGLER ADM3A (Kit) —
including RS232 Cable — when purchased
with Horizon or Cromemco System
. . . . ONLY $769

PRICES IN THIS AD ARE FOR
PREPAID CASH ORDERS. Slightly
higher prices apply for credit card
and C.O.D. purchases.

WRITE FOR FREE CATALOG on
hundreds of items at discount prices.
Credit card and COD shipments also
at discount, but slightly higher than
cash prices.

CROMEMCO
SYSTEM 3

List $5990   OUR CASH PRICE **$4999**

Features 4 MHz CPU, 32K of RAM, dual PerSci
floppy disk drive (and provision for installing two
additional drives), RS232C Interface, Printer
Interface; assembled and tested, ready to use.
(Order as 02-5503-0)

System 2 — Similar to System 3, but
   features dual minifloppies, List $3990
   (order as 02-5502-0) **$3390**

Z-2 Computer System, Kit for rack
   mounting, Z-80 processor, 21 slots,
   power supply, front cover panel
   List $595 (order as 02-5301-0) 499

   Assembled and Tested, incl. fan and
   all edge connectors, List $995
   (order as 02-5401-0) 845

Z-2D Disk Computer System, Kit
   Similar to Z-2, but comes with floppy disk
   controller, DOS, and minifloppy disk drive.
   A complete system with the addition
   of a RAM Board, List $1495 (02-5302-0) 1270

   Assembled/Tested List $2095
   (order as 02-5402-0) 1780

CROMEMCO BOARDS

4 MHz Single Card Computer,
   Kit, List $395 (order 02-3511-0) . . . . $ 335
   Assembled/Tested $450 (02-4511-0) . . . 382

Bytesaver PROM Board and PROM
   Programmer, Kit, without PROM
   List $145 (order as 02-3308-0) 123
   Assembled/Tested List $245 (02-4308-0) 208

TU-ART Digital Interface, Kit
   List $195 (order as 02-3440-0) 165
   Assembled/Tested, $295 (02-4440-0) . . . 250

Disk Controller Card, Kit
   List $395 (order as 02-3701-0) 335
   Assembled/Tested, $595 (02-4701-0) . . . 515

TV Dazzler, Kit $215 (02-3501-0) 182
   Assembled/Tested, List $350 (02-4501-0) 297

CROMEMCO DISK SOFTWARE

BASIC, Fortran, Assembler, Cobol, Word
Processing System, Data Base Management,
all complete with CMOS disk operating
system, List $95 each . . . . $80 each

Multiple User BASIC Now Available!

For shipping, add $2 for boards, $6 for floppy disk systems, $12 for Horizons, $15 for Cromemco
Z-2 and Z-2D Kits. Assembled Cromemco systems, DEC printers, and Teletype 43 units are
shipped freight collect.

Above prices subject to change without notice. All offers subject to withdrawal without notice.

STORE HOURS: 9-5 Mon-Fri, Sat & Evenings by appointment.

MiniMicroMart, Inc.

1618 James Street, Syracuse, New York 13203 (315) 422-4467

```
0105' 79          MOV    A,C     ;GET THE BYTE
0106' E66F        ANI    6F      ;SET THE STROBE
0108' D3A1        OUT    0A1     ;AND DO IT
010A' F680        ORI    80      ;  AND TURN IT OFF
010C' D3A1        OUT    0A1     ;
010E' C9          RET            ;
          .END
```

SAMPLE TABLE

```
BKSP    0049'     CARDY   00CA'   CARMV   00B9'   CARSP   009B'
CARTN   005B'     CHAR    000F'   CR      0051'   CRLF    0067'
DIABLO  0000'     LF      0070'   LINEF   0074'   LINF1   0077'
LINS2   0065'     LINSP   00A6'   PLRDY   00FF'   PLTMV   00F2'
PRN1    0024'     PRNAS   0017'   RETN    0095'   RLINF   007D'
RSETL   006A'     SPACE   0039'   SPC1    003D'   SPVAL   003F'
STORE   00EE'     SUBTR   00E7'   TENCI   00A5'
```

Hex listing of the diablo driver for location FA00

```
>dFA00 FB0E

FA00  F5 C5 D5 E5 3E E0 D3 A1 DB A0 47 79 E6 7F 4F FE
FA10  21 38 26 FE 7F 28 22 78 0F 0F E6 0C F6 F0 D3 A1
FA20  57 79 D3 A0 DB A0 CB 47 20 FA 7A E6 DE D3 A1 F6
FA30  20 D3 A1 3E E0 D3 A1 18 04 FE 20 20 0C 16 00 CD
FA40  9B FA 5F CD B9 FA C3 95 FA FE 08 20 04 16 08 18
FA50  EE FE 0D 20 12 CD 5B FA C3 95 FA 2A FC A3 7C F6
FA60  0B 57 5D CD B9 FA C9 FE 11 20 05 CD 5B FA 18 04
FA70  FE 0A 20 09 CD A8 FA CD F2 FA C3 95 FA FE 0C 20
FA80  09 CD A8 FA 7A F6 06 57 18 ED FE 14 20 07 AF 57
FA90  5F ED 53 FC A3 E1 D1 C1 F1 79 C9 78 E6 80 FE 80
FAA0  20 03 3E 0A C9 3E 0C C9 AF 57 78 E6 40 FE 40 20
FAB0  04 3E 08 5F C9 3E 10 5F C9 ED 53 F0 A3 7B 2F D3
FAC0  A0 7A 2F E6 0F F6 E0 D3 A1 4F DB A0 CB 4F 20 FA
FAD0  79 E6 AF D3 A1 F6 40 D3 A1 ED 5B F0 A3 CB 5A 20
FAE0  06 2A FC A3 19 18 07 E6 07 2A FC A3 ED 52 22 FC
FAF0  A3 C9 7B 2F D3 A0 7A 2F E6 0F F6 E0 D3 A1 4F DB
FB00  A0 CB 57 20 FA 79 E6 6F D3 A1 F6 80 D3 A1 C9
>AC=C
```

New from Echo

A complete line of polyester films (mylar), suited for the majority of your graphic art, drafting or computerized plotting needs.

Twelve sizes of cut sheets in stock. Buy one sheet or take advantage of our 100 sheet per size discount. Roll stock available in eight widths. Choose from two registration punches: Graphic punch 3/16" slot, 1/4" round, 3/16" slot on tenth inch centers. Drafting punch 1/4" slot, 1/4" round, maximum spacing 30 inches. Aluminum registration pins with poly tabs available for either type of punch at $1.00 per pin.

For more information, price sheets and samples contact Echo Design & Development Corp., 195 E. Gish Rd., San Jose, CA 95112, (408) 292-0918.

Readers who mailed an inquiry card on this new product from the February 1978 issue are asked to resubmit, as all inquiries for this company were forwarded to the wrong address.

CIRCLE INQUIRY NO. 128

CCSA-Type Switching System

The Release 5, a low-cost option of the ROLM® CBX Business Telephone System software, will enable small and medium size companies to install private CCSA-type switching systems.

ROM's network uses the standard 7-digit numbering plan for inter-office calls and 10-digits for off-net calls; it can be retrofitted into existing CNX installations.

The ROLM CCSA System can also be used for regular long distance traffic. The phone user simply dials "9" plus the number desired, including area code. ROLM CX Route Optimization takes it from there. Call queuing can be used for making off or on-net calls.

For more information contact ROLM Corporation, 4900 Old Ironside Dr., Santa Clara, CA 95050, (408) 988-2900.

CIRCLE INQUIRY NO. 111

Complementary Transistors Give High Performance

Eight new PNP and NPN transistors with 80V collector-emitter voltages, 5A continuous collector currents and operating frequencies to 70MHz, provide high-performance in power amplifier and switching circuits.

The 2N5003, 2N5005, 2N5151 and 2N5153 PNP transistors have 100V collector-base voltages, 2A continuous base currents and emitter-base voltages of 5.5V. The NPN devices, 2N5002, 2N5004, 2N5152 and 2N5154, have similar electrical ratings permitting their use in complementary-pair circuits.

In 100-999 quantities, prices are: 2N5002, $13.00 each; 2N5003, $15.00 each; 2N5004, $18.00 each; 2N5005, $25.00 each; 2N5151, $7.00 each; 2N5152, $4.25 each; 2N5153, $8.00 each and 2N5154, $5.00 each.

For more information contact Solid State Devices, Inc., 14830 Valley View Ave., La Mirada, CA 90638, (213) 921-9660.

CIRCLE INQUIRY NO. 112

Three-State A/D Converter

Teledyne Semiconductor has expanded its line of monolithic data conversion products with the addition of 8, 10, and 12 bit A/D converters with three state binary output.

The device utilizes low power CMOS technology and is fully self contained in a single 24-pin DIP requiring only passive support components. Its integrating principle of operation gives it inherently high accuracy, linearity and noise immunity. Conversion speed is 1 to 20 ms.

The device is available in plastic or ceramic packages. 100 quantity prices for the 8-bit unit in plastic, $8.95; 10-bit plastic, $11.50 and 12-bit ceramic, $25.00 Delivery is stock to four weeks. For more information contact Teledyne Semiconductor, 15840 Ventura Blvd., Encino, CA 91436, (213) 986-8506.

CIRCLE INQUIRY NO. 113

Quay 80 F1

The Quay 80 F1 is a floppy disk system for use in S-100 bus computers. The Quay 80 F1 system includes the Q/80 FDC (floppy disk controller) board capable of supporting up to four disks, QDOS disk based operating system, the Q/FD1 125 KB 5¼" band-driven disk drive with power regulator and interface cable, and the Q/80 FC floppy disk cabinet.

In addition to the floppy disk support, the Q/FDC has available a programmable 8-bit, TTL compatible, parallel I/O port capable of supporting standard peripheral devices such as line printers, tape punches, keyboards, etc.

Price for the Quay 80 F1 system is $795. Add-on drives (Q/FD1) are $395 each. Delivery is 30-60 days ARO. For more information contact Quay Corporation, P.O. Box 386, Freehold, NJ 07228, (201) 681-8700.

CIRCLE INQUIRY NO. 114

Selector Switch

The Model 8544-D, A, B, C, D, CRT Selector Switch allows the user to switch any 2-wire input to any one of four 2-wire outputs. All connections are made at the rear panel.

In application, the Model 8544-D (Desktop) module allows the user to manually select any one of four CRT displays. The unit is ideally suited for switching the IBM 3270 interface or any 2-wire telephone line.

A four-position rotary switch on the front panel instantly switches any 2-wire input from a rear panel BNC labeled Common to any one of four BNC's lebeled A to D. This module is available for desktop switching only, and no power is required.

Th Model 8544-D is priced at $160. Delivery is 30 days ARO. For more information contact Marketing Dept., International Data Sciences, Inc., 100 Nashua St., Providence, RI 02904, (401) 274-5100.

CIRCLE INQUIRY NO. 124

Intelligent Floppy Disk Controller

The IFC-8400 controller will control up to eight SA400 or SA800 Shugart single-sided, single density flexible disk drives.

The IFC-8400 will permit interface to any computer or stand-alone terminal over an RS-232C or 20 mA current loop serial channel or optional parallel 8-bit TTL I/O channels.

The IFC-8400 also includes a 1K byte RAM buffer to hold data being transferred between host and diskette. Use of a buffer RAM larger than a single sector size allows for certain commands to be implemented more efficiently than with a smaller buffer. This decreases command execution times by decreasing unnecessary head movements.

The new IFC-8400 is priced at $795 and is available immediately. For more information and quantity pricing contact Cybernetic Micro Systems, 2378A Walsh Ave., Santa Clara, CA 95050, (408) 249-9255.

CIRCLE INQUIRY NO. 125

Single Chip Microcomputer Has On-Board NMOS A/D Converter

The 8022 is a low cost, general purpose single-chip microcomputer containing a full analog-to-digital (A/D) converter. Aimed at high volume control appliaitons, the microcomputer is ideas for applications such as home appliances, test and measurement instruments, automotive, process control, environmental control, sensing/recording instruments and other control applications.

The 8022 is software compatible with other single-chip microcomputers in the MCS-48 family of microcomputers and peripheral components. Another unique feature of the 8022 is its accurate on-chip oscillator, which can be externally synchronized with a crystal or a TTL-level clock signal.

For more information contact Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051, (408) 249-8027, Rob Walker.

CIRCLE INQUIRY NO. 117

MOPS Software Packages Enhances Debugging Capabilities of COSMAC

A new software package, the Micromonitor Operating System (MOPS) CDP18S831, provides Micromonitor users with enhanced debugging techniques ranging from simple terminal-Micromonitor dialog to hands-off system testing with commands coming from disk files.

The MOPS CDP18S831 consits of a MOPS diskette plus a UART module, and connecting cable to interface the Micromonitor to the CDS. Literature support includes the *Micromonitor Operating System (MOPS) CDP18S831 Users' Guide*, MPM-231, which describes the installation and startup of MOPS, the specific commands available to the user and command usage.

In single quantities, the RCA COSMAC Micromonitor Operating System is priced at $350 and the Micromonitor at $1600. For more information contact RCA Solid State Div., Box 3200, Somerville, NJ 08876, (201) 685-6380, MOS Product Marketing.

CIRCLE INQUIRY NO. 118

Zilog Z80 Compatible DAC Card

Completely compatible with the Zilog Z80 microcomputer board this digital to analog converter card offers four or eight channels of conversion, each channel having 12 bit resolution; less than 0.5 bit nonlinearity; individual zero and full-scale adjustments; five user selectable output voltage levels; optional 4-20 ma current output; memory mapped address selection; and double buffering.

The Model 608 DAC card price is $595 for 4 channels and $895 for 8 channels, in small quantities. For more information contact Signal Laboratories, Inc., 202 N. State College Blvd., Orange, CA 92668, (714) 634-1533, Del Flagg.

CIRCLE INQUIRY NO. 126

Two New Printer Interfaces

Sol Hytype I mounts inside any Diablo Series 1200 Printer connecting it directly to the back of the Sol. Similarly the Sol Hytype II Printer Interface works with the Diablo Series 1300 Printer.

The installation package includes the fully assembled, tested and burned-in printed cir-

cuit board, software, all cables and mounting hardware. No modification to the Sol is necessary. No holes need be drilled in the printer. The printer can be restored to its original condition if required.

Hytype driver software is included on CUTS cassette along with a source listing. The user may modify the driver software to suite a particular application.

Suggested retail price for both the Hytype I and Hytype II is $150. Delivery is stock to 30 days. For more information see your Sol dealer ad contact Processor Technology Corp., 7100 Johnson Industrial Dr., Pleasanton, CA 94566, (415) 829-2600.

16-Bit Microprocessor Family

The MCS-86™ Microprocessor Family is comprised of the 16-bit 8086 CPU and its peripheral support components, development software, and design development aids. Designed to delivery ten times the performance of the 8080, the 8086 provides features never before found on a microprocessor.

Some of these features are extended addressing capability — up to one million bytes; 16-bit hardware multiply/divide; elaborate string handling instructions; dynamic memory relocation; reentrant program code, position-independent programs; instruction look ahead.

These features, while providing a new architecture, at the same time maintain compatibility with the 8080 and 8085 microprocessor families. For more information contact Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051, (408) 249-8027, Rob Walker.

MCZ-1 Microcomputer Implements New Integrated Terminal

The MCZ-1/60 consists of an intelligent terminal with 4K bytes of RAM expandable to 52K bytes, a general purpose computer with 32K bytes of RAM expandable to 64K bytes, and an integral 9-inch CRT — all in a single desk-top unit. Dual rack-mounted floppy disk drives provide 600K bytes of on-line program and data storage.

Incorporating Zilog's own Z80-CPU microprocessor and 16K dynamic RAM chips, the MCZ-1/60 sells for $6,990 in single quantities with delivery in 60 days. The system is also available in a version with desk-top disk drives, designated MCZ-1/62. For more information contact Zilog, 10340 Bubb Rd., Cupertino, CA 95014, (408) 446-4666, Dave West.

Graphic/Text Video Interface

Merlin is a combination text and graphic video display board, combining functins of text display, graphic display (320H by 200V resolution), keyboard input port, and 4K bytes of on-board control ROM.

Merlin displays 20 lines of easily readable text with 40 characters per line. This is suitable for text editing, BASIC and assembly programs and large screen classroom use. Both upper and lower case characters can be displayed.

The Merlin video interface provides the main console I/O in a small system, or can be the heart of a sophisticated graphic development system.

Price for Merlin assembled and tested is under $500. In kit form without ROM software price is less than $300. For more information contact Miniterm Assoc., Inc., Dundee Park, Andover, MA 01810, (617) 470-0525, Dave King.

Digital Logic Modules

The Wyle line of digital logic includes over 200 modules covering all types of logic elements. Available on the 3¼" x 4½" modules are gates, flip-flops, decoders, counters, one shots, line drivers/receivers, electronic and many more.

Apple II is at The Computer Store

The Apple® II, today's most popular personal computer, is at The Computer Store. Along with the latest in Apple peripherals. Like the new Disk™ II floppy disk drive. Or, printer and communications interfaces. And, the latest in software including the new Apple/Dow Jones Stock Quote Reporter. The compact Apple II gives you 48K RAM memory with full color graphics and high resolution graphics. It's the most powerful computer in its price range.

At The Computer Store, we have more than ever before in microcomputers, memories, terminals and peripherals. All backed by a technical staff and a full service department. Stop in today, you'll find more than ever before at The Computer Store.

The Computer Store
820 Broadway, Santa Monica, California 90401 (213) 451-0713
The Original Name In Personal Computer Stores
Store Hours: Tues.–Fri., Noon–8pm, Saturday, 10am–6pm
Located two blocks north of the Santa Monica Freeway at the Lincoln Blvd. exit.
Phone and mail orders invited. BankAmericard/Visa and Master Charge accepted.

SPACE AGE COMPUTER STORE
New York City and Long Island

*If You Want Professional Service
In A Casual Atmosphere –
And A Large Variety of Equipment*

BUSINESS APPLICATIONS
General Ledger, Inventory, Accounts Payable, Receivable, Word Processing

GENERAL APPLICATIONS
Northstar Mailing Label Program . . . $45

Northstar Macro Assembler . . . $65

STOCK MARKET PACKAGE — (Unique)
Makes Ticker-Tape Obsolete
Send $2.00 For Descriptive Brochure And Much More

BYTE SHOP
the affordable computer store

130 East 40th Street	2721 Hempstead Turnpike
New York, NY 10016	Levittown, NY 11756
(212) 889-4204	(516) 731-8116
(corner Lexington Ave.)	(Just E. of Wantagh Pkway.)

11-7	Tues, Thru Fri	12 to 8
10-5	Saturday	10 to 5

Additional modules include relays, test point modules, extender, lamp, toggle switch and a wide variety of socket, wire wrap and blank modules. Also available are card fiels and card drawers for rack mount or custom installations, and logic power supplies.

For additional information contact Wyle Laboratories/Computer Products, 3200 Magruder Blvd., Hampton, VA 23666, (804) 838-0122.

Air Velocity Detector Increases Reliability of Air Cooled Equipment

The Dietz Model AD Air Velocity Detector will indicate air flow in the range of 1 foot per second to an infinite air velocity. It can be used to detect the lack of air flow in air cooled electronic equipment, such as computers.

The Dietz Model AD Air Velocity Detector provides a sensitivity to air flow that increases as the air flow diminishes, is immune to shock, vibration, and ambient temperature changes, is low cost, and of small physical size.

The Model AD can be mounted across the face of the blower, requiring only an additional ⅜" of space. For additional details contact Henry G. Dietz Co., Inc., 14-26 28th Ave., Long Island City, NY 11102, (212) 726-3347, Henry Dietz, president.

Literature on Subminiature Switches

C&K Components, INc., is now offering its latests addition to its line of consumer-oriented literature. SFC 678 includes 16 pages of subminiature and microminiature toggles, rockers, pushbuttons and power switches from the extensive C&K switch line.

Complete with model numbers, specifications, P.C. layours, mounting information and prices, this catalog will fill the needs of every OEM and designer. For more information contact C&K Components, Inc., 103 Morse St., Watertown, MA 02172, (617) 926-0800.

INTRODUCTION

to the

TEX AND REAL APPLICATIONS

Now you'll see some examples from a system for physical mail (electronic mail is interconnected, but not shown here). Many readers will find it useful to study the programs for the algorithms, if not for TEX. Their challenge is to write the same programs in BASIC or any other language, for comparison with these TEX programs for the same procedures. I'll guarantee that TEX is much easier and shorter!

The mail system has these parts (individual programs):

- Creating the name/address file.
- Updating the name/address file.
- Adding or changing indication of membership on a specific mailing list.
- Displaying a mailing list, on a cover sheet or as labels.
- Archiving the various source lists; that is, making an integrated set of permanent copies.

The original database is the telephone directory. People working for a company often consider the accuracy of their entry in it to have priority second only to payroll. To understand the programs to be explained here, the database format needs to be known. It is a linear/sequential file of entries of this form:

```
surname, (given) # tel-no # org. # address # room # bldg
```

The entries are of variable length. "address" is a 4-character mail-station here, but it could be a full address just as easily. "#" is used to represent visibly the Horizontal Tab character. It gets replaced automatically in all display programs.

LINE PRINTER DISPLAY OF THE TELEPHONE BOOK

First let's look at a program called "alphatel", which is the proofing run before photocomposition of the telephone book (directory).

```
!alphatel clear * restart=0                                      a
!restart0 out:"`alphatel' has restart capability."              b
filename="telbook,q" call texlib/old if fail call ouch          c
out:*lf,"Have the file." rs:"#";*:*ht b out:"Tabs in."          d
call texlib/datehead b ib:*cl:".pape 65"                        e
i:*cl:".repl ^" i:*cl:".tabu 27,37,55,61,69"                    f
filename="sink" call texlib/resa restart=1                      g
out:"Restart1 passed. Now you can, in case of"                  h
out:"any failure, restart at the last restart"                  i
out:"passed by entering `call alphatel!restart'."               j
goto !nostart1                                                  k

!restart1 filename="sink" call texlib/old f:".tab"              l
!nostart1 i:*cl:"*hd" i:*cl:".space" i:*cl:".space"            m
call separate call widow                                        n
out:"Widowing operation complete",*lf                          o
b r:"*hd";*                                                     p
                    Organization/  Mail                         q
.break                                                          r
    Name    Phone    Component    Sta.   Room   Bldg.          s
.break                                                          t
                                                                u
.space                                                          v
*null                                                           w
b out:"Restart2 (paging) passed." call texlib/resa              x
restart=2 goto !nostart2                                        y

!restart2 filename="sink" call texlib/old                       z
!nostart2 call texlib/formsink which="a" name="sink"           aa
id="(my-ID)" bill="(my-charge)"                                bb
call texlib/print !howmany nosubs return                       cc

!restart subs $                                                 dd
goto !restart$restart$                                         ee
```

Figure 1. The program "alphatel"

In Figure 1, the lines of "alphatel" to be explained are keyed to letters on the right:

b   Programs that use a fair amount of computer time should have restart capability, so that not all of the work is lost in case of a failure. The user of this program is advised that it is restartable.

c   At the terminal we would just say "old telbook" to get that file. If it was busy, or had a password we

Part Three

TEX *Language*

By Robert W. Bemer

did not provide, a message to that effect would be displayed. Manual correction action could be taken. In executive files such actions must be automatic. So we set the content of the variable "filename" to be the name of the wanted file, and call a program named "old". But this isn't one of our personal files. It belongs to another (but synthetic) user called "texlib". All of its programs constitute a library of service routines and general applications. "old" contains:

```
subs | if filename:eqs:*null goto !fail
ergo !fail old |filename|
fail='f' |*svmd| return
!fail fail='t' |*svmd| return
```

The error forms for "goto" and "call" verbs are "ergoto" (or "ergo") and "ercall" (or "erca"). They are obeyed only if anything following them (on the same line) fails. This permits "failsoft" operation and recovery from failures. In this case a failure to get the requested file sets the variable "fail" to "t", and returns.

A comma and the letter "q", for "query", follow the name of the wanted file. This gets us a snapshot copy of the file even if someone else is then changing it. If even this should somehow fail, "ouch" is called to repair the situation.

d   In any lengthy process, it's sensible to advise the user (even yourself) how it is going. All "#" characters are replaced by Horizontal Tab:

"rs:" means replace the string
";" means do it the following number of times
"*" means all
":" means with the character following

When the process reaches end-of-file, "b" means back up to the beginning. After this is done, another advisory message.

e   The library program "datehead" prefaces the current file with a message "This display requested by (your name) on (the date) at (the time)". It's always useful to do this, to distinguish among several versions or runs. "ib" stands for "insert

before" (the current line) a line that has the content enclosed in the delimiters.

f   After that line (but still before the first current line), two more lines are inserted. One tells the formatting program to replace all characters "^" with incompressible spaces; the next gives the tab stop positions.

g   Now the content of "filename" is changed to "sink", which is where we wish to keep the intermediate results of our process. The library program "resa" (for resave) puts the current file into "sink", and the variable "restart" is set to 1.

h   Now we can tell the user that he can restart, and how to do it. Suppose something goes wrong just after this point. Per instructions the user would enter "call alphatel!restart".

dd   The program would recommence execution at the label "restart", which sets the substitution (subs) mode with the $ sign. We can't continue on this same line because all substitution in a line is done before any execution of the line begins. So the subs mode must always be set one line prior to its usage.

ee   Now the value of "restart" gets substituted, and in this case we go to line "l".

k   Line "l" is skipped, and control is at line "m".

l   If we had to restart, the correct file position is located by finding the line starting with ".tab".

m   Three lines are inserted. One is a dummy to stand for the eventual heading, and two cause spaces in the display program.

n   "separate" separates the last name starting with "A" from the first starting with "B", etc., and puts the starting letters in the gaps. "widow" does the paging, ensuring at the same time that no new initial letter group starts unless there are at least three entries in that group before a new page begins.

o   Another "we're still here and working" message.

p   The dummy heading "*hd" is replaced by the real one, which is specified in lines "p" through "v". The ".break" commands force new lines.

u   This yields a continuous underline in photocomposition.

w   This simulates the extra Return to show that the replacement is complete.

x   Back to beginning of file, which is put into "sink" in the new and modified form.

y   The restart count is bumped up, and control goes to line "aa".

aa   "formsink" automatically formats the file and puts it in "sink". The ASCII line printer routine is set by "a", and the file to print is identified.

bb   "id" specifies *whose* file "sink", and "bill" is for timesharing charges.

cc   The only variable left unspecified is "howmany", so the "print" program is called at that entry point. After that starts, the subs mode is turned off, and the program ends.

MAKING THE ORIGINAL MAIL CONTROL FILE

The goal is to create a new file (called "telmail") by extracting the name and address from each line entry. Only programs (not people) access this file for modification, so the "#" character is unnecessary. The file entry format is:

```
name HT address\.1.16.5.23.38.14.
```

This means that this individual's name occurs on special mailing lists 1, 16, 5, 23, 38, and 14. The periods delimit the list numbers uniquely. But this is after "telmail" has been processed many times. The original format is:

```
name HT address\.
```

Figure 2 is the program that builds "telmail" originally.

```
!makemail out:*lf,"Starting 'makemail' at ",*time,*lf    a
clear * call !setup ht=*ht call !fixup                    b

!nameit ib:*cl:("This file created ",*date) |official|    c
!nameit_again |suffix|                                     d
filename=sinkname |make_new_else| goto !nameit_again      e
|exit1|                                                   f
out:"'",sinkname,"' contain name and address" |exit2|     g

!fixup filename="telbook,q" |get_it|                      h
out:"Have the file" d:"^^^";* b                           i
!lineLoop ro:"#";3:*rs scan:*cl:*rs                       j
r:*cl:(*l'>"#",ht,*r'>"#","\.") f;1 |eof|                  k
goto !lineLoop                                            l

!setup cant=\b out:"Can't find ",_name goto !match\       m
bump=\count=count+1 name="n",count\ t=\in:"List number\   n
what_no=t,\is?  " tag=*in if (*in>`*n):ne:*lin goto\      o
official=\out:"For the official 'telmail', just CR"\      p
t=\in:"Else what suffix?  " sinkname=*userid,"/telmail"\  q
suffix=t,\,*in if *lin:eq:0 sinkname="the/telmail"\       r
exit2=\nocase nosubs out:*lf,"Done at ",*time return\     s
exit1=\if fail out:*lf,"No action is taken."\,exit2       t
if_yes=\if *in'J1:eqs:"Y" \                               u
retry=\in:"Try another suffix?  " \,if_yes                v
t=\if fail out:"A file '",sinkname,"' \                   w
doesnt=t,\does not exist." \ does=t,\already exists." \   x
get_it=\call texlib/old \ make_new=\call texlib/save \    y
get_it_else=get_it,doesnt,retry                           z
make_new_else=make_new,does,retry                         aa
null_cf=\call texlib/new\ put_away=\call texlib/resa\     bb
eof=\if *eof b return\ case subs | return                 cc
```

Figure 2. The program "makemail"

In explanation of Figure 2:

b   The subprogram "setup" (lines "m" through "c") defines variables to have certain procedures as content. Used by most of the component programs in the mail system, they are explained as encountered. They compact the programs and make them easier to read and to understand.

   The variable "ht" is defined to be the Horizontal Tab character. The program is called at the label "fixup".

h   "telbook" is gotten as the current file, again on a snapshot basis. "get_it" is a procedure. For a different computer we would redefine "get_it" to be the corresponding procedure. Thus the kernel of the mail program is portable.

i   After a "working" message, all lines starting "^^^" are deleted. They are the redundant entries for secretaries listed following the people they work with. Back to beginning of file.

j   A loop operating on all remaining lines. "ro" means "replace occurrence". So the third occurrence of the "#" character is replaced by a Record Separator character. The current line is then scanned on that character.

k   Both left and right parts are kept only as far as the first "#" encountered, and adjoined by the content of "ht", which was a HT character. The two characters "\." are placed at string end, and the entire string replaces the original line. The pointer is moved to the following line. If an end-of-file signal is received, the process returns to line "c".

l   Else it's repeated until the entire file is converted.

c   For the record, an identifying line is inserted automatically at the beginning of file. Then the user is asked if it's the official telmail file he's making.

d   If he replies with just a CR to "Else what suffix?", it will be the official file. Else it will be a personal copy for private purposes. E.g., for a list of the rockhounds, or equestrians. That file name is the contant of "sinkname".

e   The file this program will make should not exist yet. So an attempt to create it is made. If it fails, a message says that a file of that name already exists, and would you like to change your mind about the suffix?

f   If the user gives up, the creation attempt was still a failure, and the program closes by saying that no action was taken.

g   With success, we are so notified, and the process is wrapped up and complete.

UPDATING THE MAIL CONTROL FILE

People leave, and new people come. Those still there get transferred or moved to new offices. And so the content of the telephone directory changes — particularly the address (mail station in this case). Periodic updates of two types are desirable:

• Frequent microfiche copies, or online interrogation, for the switchboard operator.

• Less frequent photocomposed and published copies for all employees.

The programs of this article were motivated in part by failures to get mailing lists changed or corrected by human procedures, often even after two years. And when I converted the old punch card system for the telephone directory, the process turned up six deceased, a woman who had left four years ago for motherhood, and a fictitious "Fred Fortran" in Manufacturing!

Figure 3 shows the program "update", which updates the file "telmail" for currency and correct address content, while preserving the existing mailing list membership data.

```
!update out:*lf,"Starting 'update' at ",*time,*lf        a
clear * call !setup |official|                            b
!update_again |suffix|                                    c
ht=*ht,*us call !fixup                                    d
 filename=*;sinkname |get_it_else| goto !update_again     e
 |exit1|                                                  f
out:"Have both files" sort *;*:(A25)(A1)                  g
out:"These names are new, in case you"                    h
out:" wish to add them to any list.",*lf f;1              i
call !just1 b rs:ht;*:*ht filename=sinkname |put_away|     j
call !listold out:"'",sinkname,"' updated" |exit2|        k

!just1 |eof|                                              l
scan:*cl:ht if *lm:eq:0 cut;1 goto !just1                 m
!new nname=*l f;1 scan:*cl:ht                             n
if *lm:ne:0 out:nname goto !new                           o
if nname:eqs:*l scan:*r:"\." d b;1 a:*r f;1 goto !just1   p
cut;1 out:nname goto !just1                               q

!listold |null_cf| b paste b a;*:"|" b cuts:"\.|";* b    r
out:"These names are no longer valid."                    s
out:" Rerun indicated lists to remove them.",*lf          t
p;* paste return                                          u
```

Figure 3. The program "update"

In explanation of Figure 3:

b   With line "c", the same procedures used in making the original telmail file.

d   Now the same creation process ("fixup") is applied, except that this time the variable "ht" has the extra Unit Separator in it.

e   The filename is defined to adjoin the new file just made with the old file. We wish to transfer the mailing list identifications from the old file to the new, scrapping the old entries. Again, existence is controlled by the procedures. If we wish to give up, that happens via the "exit1" procedure in line "f".

g   A "working" notice. The sort procedure is called. The Unit Separator, as a control character, collates low to any graphic. Thus when the same person appears in both new and old files, the new entry precedes when the two files are sorted together.

The sort verb reads "Sort the current file (*) into the current file (i.e., in situ); an alphabetic field of the first 25 characters is defined as the first sort key; sort ascending (A) on the first field".

h   With line "i", a signal that there may be names in the telephone directory now that weren't in there before.

j   The subprogram "just1" is called to ensure that double entries have the information transferred to the new entry, and to delete old entries without a corresponding new.

l   If end-of-file, return.

m   Scan for "ht", which still contains the Unit Separator. If the length of *middle is zero, we didn't find it, so the line is an OLD entry. In that case we cut it from the file (it is added to a separate "cut" file, and thus deleted from the current file) and go back to try the next line for a new entry.

n   If we get here we have a NEW entry. "nname" is the new name. The next line is inspected to see if it is a matching OLD entry.

o   If "ht" is found it's a NEW, not OLD, entry. So the previous entry was a new listing in the phone book, and "nname" is printed out according to the

cover message of lines "h" and "i". Back to try again for a pair.

p   Now there is a NEW-OLD pair. Do they match? If so, the tag information is picked up from the OLD entry as *r. The OLD entry is deleted; a backup of one line points to the NEW entry. "a:*r" means put *r after the text of the current line. That entry is now fixed, and we go to the next line to repeat the process.

q   If we get here, it's a peculiar coincidence that a new person and one no longer in the directory just happen to be adjacent in the ordering. The OLD one is cut, and the new one printed. Upon end-of-file we go back to finish line "j".

j   Back to beginning of file, which now contains only NEW entries. "ht" is replaced in each line by HT,

---

TEX is. . .useful for prototyping applications that might eventually be done in compiler-type languages. It checks out design and human interfaces fast. Most debugging is done. . .on live data.

---

and the file is resaved.

k   Having listed all new (added) names during the process, in case they should go on any mailing lists, we call the subprogram "listold", to print the names to be removed from the source mailing lists.

r   A null current file is made per line "cc" of Figure 2. The cut file is pasted to it. After every line we put a vertical bar character, and back to beginning. Now all lines containing the string "\.|" are cut, because if such a string is found the entry is not on any list, so why bother to tell anyone? And to the beginning again.

s   With line "t", tells the user what will be listed.

u   "p;*" means "print all lines". The cut file is pasted just to clear it out for future processes. Return to line "k" and wrap-up of the process.

```
!chnglist out:*lf,"Changing list at ",*time,*lf goto !sk a
!newlist out:*lf,"Adding list at ",*time,*lf             b
!sk clear * call !setup                                   c
!list_no |what_no| !list_no                               d
filename="mail",tag |get_it| count=0 p;2 f;1              e
call !vector |official|                                   f
!suf |suffix|                                             g
filename=sinkname |get_it_else| goto !suf                 h
 |exit1|                                                  i
rs:(".",tag,".");*:"." b count=0 call !match |put_away|   j
out:"'",sinkname,"' updated by 'mail',tag,"'" |exit2|    k

!vector if ^*eof |bump| _name=*cl'<^" " f;1 goto !vector l
limit=count out:"'mail",tag,"' names vectored." return    m

!match if limit:eq:count return                           n
|bump| scan: name:"," if *lm:eq:0 call !exception         o
last=*l,*m f:last if *eof b f:last if *eof |cant|          p
call !components M="A"                                     q
W1=W3 A1=A3 L1=L3 W2=W4 A2=A4 L2=L4 max=0 linect=0         r
!lastloop scan:*cl:","                                    s
if (*l,*m):nes:last call !high goto !match                t
```

Figure 4. Branched to next page

```
Figure 4.  Vectored from previous page

scan:*r:*ht split:*l:0 call !components n=0                          u
if L2:ne:0 if L4:ne:0 call !region1 goto !have_it                    v
if L2:eq:0 if L4:eq:0 call !region4 goto !have_it                    w
!region23 del=1 M="A" if L1:gt:1 if L3:gt:1 del=3 M="W"              x
if |M|1:eqs:|M|3 n=del if n:eq:3 goto !have_it                       y
if L2:eq:0 W2=W1 A2=A1 W3=W4 A3=A4                                   z
del=1 if L2:gt:1 if L3:gt:1 del=3 M="W"                             aa
if |M|2:eqs:|M|3 if del:gt:n n=del                                  bb

!have_it n=n*1000+900-linect if n:gt:max max=n                      cc
if ("000",max)'[3:eq:6 call !high goto !match                       dd
linect=linect+1 f;1 goto !lastloop                                  ee

!high t=(900-max['3) if max:lt:1000 b;t |cant| return               ff
b f:last f;t a:(tag,".") b;t return                                 gg

!region1 del=1 if L2:gt:1 if L4:gt:1 del=3 M="W"                    hh
if |M|2:nes:|M|4 return                                             ii
n=del M="A"                                                         jj

!region4 del=1 if L1:gt:1 if L3:gt:1 del=3 M="W"                    kk
if |M|1:nes:|M|3 n=0 return                                         ll
n=n+del return                                                     mm

!components scann:*r:" " split:*r:1 A3=*l scan:*r:*uc               nn
W3=(A3,*l)'<*a A4=*r'J1 W4=*r'<*a L3=W3>*sub L4=W4>*sub             oo
if L3:gt:1 scan:W3:" " if *lm:ne:0 call !exception                 pp
if L4:gt:1 scan:W4:" " if *lm:ne:0 call !exception                 qq
return                                                             rr
```

Figure 4. The program "newlist"

SETTING UP A NEW MAILING LIST

Figure 4 is the program "newlist", used to add indications for a new mailing list, and to update an existing but changed list. Refinements could be made for faster operation and more elegant decisions for search terminations. But that gets too large to show here!

In explanation of Figure 4:

c   With line "d", the standard procedure to begin.

d   "what_no" (Figure 2, line "o"), permits no reply except digits. It does so by scanning the input for a character not a digit. If the count is less than total length, it's not all digits.

e   The designated mailing list is brought as the current file. The first two lines (identifying data) are printed. Then we move to the first name to find.

f   The subprogram "vector" puts each name of the mailing list into one variable of a vector, the names of which run from "n1" to "ni".

l   If not done, "bump" (Figure 2, line "n") ups the subscript count 1 and makes the content of "name" the current line with any righthand spaces removed. The process loops on this single line until end-of-file.

m   When the whole file is assigned, the last value of "count" is assigned to "limit", so the stepping process can be reconstructed. A message, and return.

g   With lines "h" and "i", a familiar process.

j   The old indicator for this mailing list (number and bracketing periods) is replaced by a single period everywhere it occurs. If it's a new list being added, none are found. The count is reinitialized to use the name vector in the matching process, and it is called.

n   If the count is at limit, the name vector is exhausted. Return to line "j", save the modified file, wrap-up.

o   The name string is broken on the comma, which is critical in two ways: 1) If there isn't any we have a problem, and must execute "exception" (not shown).

p   2) The comma must be included in "last" for the search, else we might think that "Johns" was found when it was actually "Johnson".

If not found, backup to file beginning, in case the file may be in wrong sequence. But if such a last name still isn't found, the "cant" procedure (Figure 2, line "m") is executed. It takes the program back to the label "match", for the next name.

q   "components" is called to obtain the first and middle names or initials. "M" is set to "A", which identifies the name of initials. Whole names use "W".

nn   *right, including all but the last name, is scanned for *not* space. This handles 0 to n spaces after a comma.

W3 and W4 the first and middle names. L3 and L4 will be their lengths in characters, and A3 and A4 the initials of those names.

The first initial *must* be the first character. It could be followed by another capital, a period, a space, or small letters of the complete name (which could also have a period, like "Jas." and "Jno.").

"A3" is the beginning capital of the first name. *right is broken for *its* *right to begin with a capital.

oo   The initial and residual up to the middle name are scanned from the right for the first letter, removing blanks or other punctuation between the given names. "A4" is the beginning capital of the middle name, and the middle name is found in the same way.

pp   If not an initial, the first name is checked. If it contains a space the exception subprogram is called. Line "qq" does the same for the middle name.

rr   Back to complete line "q".

r   "components" works on names in the address file, where the "3" and "4" subscripts are used. In this case it is serving a vectored name, and the subscripts are changed to "1" and "2". "linect", set to 0, is the count (from the first occurrence of the surname) where the best match is found.

"max" is the highest "n" found for any set of given names. *RULE:* If both names are present, count 3 for a full name match, 1 for an initial match, for each name (maximum of 6 is possible). If one or the other middle name is missing, match the existing middle name against the other first name if the first match fails.

s   A loop to pick up the surname in the list "telmail".

t   If no match, the set of wanted surnames is exhausted. Do the subprogram "high" to pick the best fit so far. Then get the next vectored name.

u   The address is stripped, and the given names subjected to "components" after the value is initialized.

v   If both middle names/initials exist, "region1" is called to match them. It continues into "region4" to match first names/initials.

w   If only first names for both, "region4".

x   "del" is the scoring value. It's 1 except if both names are not initials, in which case it's 3, and we compare whole names (W) instead of initials (A).

y   The test is made. If pass, "n" is set to the value. If it is 3, the first names had a full match, and there is no use checking crosswise to a middle name. E.g., "Quitecontrary, Mary Mary".

z   We must match first against the existing middle. This line effects interchange for one of the two conditions, so that lines "aa" and "bb" work for both cases.

aa   The same value setting, initial or full name.

bb   If a match, and a higher value than we have so far, the higher is it. E.g., matching "A. Andrew" to "Andrew", the middle name is a better match.

cc We get here by falling through for regions 2 and 3, by a "goto" from regions 1 and 4. The value is put in the 4th position by multiplying by 1000; the line count is decremented from 900 (in this case the maximum of a single surname that can be tested. It is decremented so that the first individual with the highest value is the match. If "n" is higher than the old maximum, it becomes the new maximum.

dd 6 is the highest possible value. Stop looking.

ee Else up the line count, go forward to the next name to test, and repeat.

ff "t" is the line count for the maximum. But if we didn't get a value of at least 1, there is no match at all.

gg To beginning of file again. Find the surname once more, and move forward (count) lines. After that entry append the list number and another period. Back up to the first of that surname and retry.

MAKING THE MAILING LIST TO USE

Figure 5 is a program "display", which makes a cover sheet, or labels, as the directing medium for a specific list.

```
!display out:*lf,"Starting `display' at ",*time,*lf    a
clear * call !setup |official|                          b
!display_again |suffix|                                 c
filename=sinkname |get_it_else| goto !display_again      d
|exit1|                                                 e
!list2 |what_no| !list2                                 f
cuts:(".",tag,".");* |null_cf|                           g
b paste b d;1 a;*:*rs b ds:"\",*rs;* b                   h
in:"Want labels?  " |if_yes| call !labels |exit2|       i
call !cover |exit2|                                     j

!cover in:"Want in mail station order?  "              k
|if_yes| call !ms_order sort *;*:(A4)(A1)               l
if *in']1:nes:"Y" call !no_order                        m
ib:*cl:("Mailing List No. ",tag) i:*cl:"  "            n
b p="n" cols=2 n=50 see="n" call texlib/n-up!bf         o
filename="sink" |put_away|                              p
out:"List is in your file `sink' "                     q
out:"You may use `texlib/2print sink(n)', or"           r
in:"list at the terminal?  " |if_yes| b p;*            s
return                                                 t

!ms_order |eof|                                        u
scan:*cl:*ht pre=*l split:*r:1                          v
r:*cl:(*l,("  ",*r)['3," ",pre) f;1 goto !ms_order     w

!no_order |eof|                                        x
scan:*cl:*ht r:*cl:((*l,"            ")']25,*r          y
f;1 goto !no_order                                     z
```

Figure 5. The program "display"

In explanation of Figure 5:

a Through line "f", standard practices of the previous programs.

g All lines containing the indication for the wanted mailing list are cut from the current file. A null file is created.

h The cut lines are pasted, and the first line of the file (a dummy blank) is deleted. A Record Separator (*rs) is put after each line. Then all strings between "\" and RS are deleted. This destroys all of the list indicators, leaving only name and address.

i if labels are wanted, that program (not shown) is called, and the process ended.

j Else the subprogram "cover" is called.

k An option is given.

l If accepted, the subprogram "ms_order" is called.

u At end-of-file, a return upon process completion.

v Else the line is broken on the HT character. The single letter prefix (in this particular scheme) is split off.

w The line is reconstructed and replaced. Now the address is in front, the numeric part of the address is right-justified, and the names are all left-justified. Go to the next line, and repeat until end-of-file.

m If the option wasn't accepted, the lines must still be conditioned.

y The lines are split. Blank spaces are inserted and truncated so that the addresses will be left-justified.

z To the next line, and repeat until done.

n The identification is put into the first line of the list, followed by a blank line.

o Four parameter values are present for the general-purpose program "n-up", so that it may be called at the label "bf" (for "brief"), and thus avoid interactive questioning. The value "n" for "see" means that we do not wish to see the process working. The name list is formed into two columns of 50 lines per page.

p The formatted list is put into "sink".

q We are told that.

r And how to print it with the line printer if desired.

s Else we have the option to print it at the terminal.

ADVANTAGES OF TEX

The user gets many bonuses from using TEX, many of which are not always expected:

• Application programs are compact. Most fit on a single page, in one field of view. This aids the human mind in comprehension. No thumbing back and forth.

• That means they also consume less storage.

• And there is less to change when you want to modify a program.

Our Dick Petersen made a database entry program and showed it to the people that had to do the work every day. Each suggested redesign to their preferences, plus some needs Dick had not foreseen. He was back in an hour, with the program running their way (a service few users get from programmers that use other languages)! Naturally they were disposed to use it. And their productivity immediately more than doubled!

• Plan-ahead and structured programming, although surely desirable, are not so vital. With TEX, I usually find it easier to jump right in and build a part, thinking meanwhile about the whole. I can always change both programs and file formats with very little effort, if that appears necessary. In the jargon of the software engineers, one can move easily back and forth between "bottom-up" and "top-down" methods.

• Programs are easy to cannibalize for other, but related, purposes.

• TEX is also useful for prototyping applications that might eventually be done in compiler-type languages. It checks out design and human interfaces fast.

• Responses to data entry prompts are easy to validate for type, size, etc. And it's simple to human-engineer a helpful request for re-entry.

• Most debugging is done quickly on live data, rather than waiting for a lengthy compilation process with vast output.

In ending this series on TEX, I'd like to say that I have been a programmer for over 29 years, and I've never felt before that so much problem-solution power was available for me to use so easily.□

GP Monitor for M6800

The GP Monitor Ver 2.1 written for the Motorola M6800 uses only the INEEE (EIAC$_{16}$) and OUTEE (EIDI$_{16}$) Software UART subroutines in the MC6830 L7 ROM. These vectors can be changed as required to any input routine and output routine that does not alter the B or X registers. The input routine must mask Bit 7 of the A register.

e.g.:

        AND A #%01111111
        RTS

The Monitor is tucked away at the very top of contiguous RAM in each version. Four versions are supplied:

        4K, 8K, 16K, 32K

The Monitor occupies the uppermost 4C1$_{16}$ bytes in each version and is protected against accidental alteration by the Monitor subroutines (the DANGER subroutine is used extensively).

The result is a virtually fool-proof general purpose Monitor useful for a variety of applications.

The Monitor accepts 18 commands from an ASCII terminal connected to an M6800 system using the MCM6830 L7 MIKBUG ROM.

When resident, the Monitor accepts input commands described as follows:

The CP Monitor has two modes of operation:
COMMAND MODE / EXECUTION MODE

The user may input two-character mnemonics which causes the Monitor to perform the selected operation. Once the operation has been performed, the Monitor re-enters command mode. All command mnemonics are followed by a comma delimiter. All address blocks are separated by commas.

e.g.:

        MM,0000,03FF,1000

Valid commands are:

LD	Load Data	TM	Test Memory
DD	Dump Data	LT	Load a Tape
LM	Load Memory	PT	Punch a Tape
DM	Dump Memory	PB	Punch a BNPF Tape
SB	Search for a Byte	CO	Calculate HEX Offsets
SW	Search for a Word	MM	Move Memory Block
CS	Call a Subroutine	CM	Clear Memory
CA	Convert ASCII	(ESC,ESC)	Go To Alternate Monitor
P1	Call Program #1	P2	Call Program #2

LOAD DATA INTO MEMORY

        LD,DDDD   D is Destination Start Address

Load data as input on terminal and store into memory starting at location DDDD. Each input increments storage pointer. If data is not stored into RAM (e.g. ROM), routine ABORTS and Command Mode are reentered.

To terminate a string of data, press ESCape key. Terminal responds by printing AAAA BB, where A is next available storage address and B is HEX count of characters entered (up to FF(255)).

DUMP ASCII DATA FROM MEMORY

        DD,SSSS   S is start address of dump

Dump data from memory to terminal starting at location S and ending when an EOT (04$_{16}$) is encountered. When EOT occurs, Monitor returns to command mode.

LOAD MEMORY WITH HEX DATA

        LM,SSSS   S is start address of storage

        e.g.: SSSS = 0000

Terminal responds by:

        0010,   NEXT LINE (16 Bytes)
            Enter comma, etc.

Whenever an exit is desired, hit ESC key and Monitor returns to command mode.

DUMP MEMORY HEX DATA

        DM,SSSS,EEEE   S - Starting Address
                            E - Ending Address

Terminal responds with:

        0000__11__22__33 ---- 00
        0010__11__22   ETC

until Ending address is reached and Monitor returns to command mode.

SEARCH FOR AN 8 BIT BYTE IN MEMORY

        SB,SSSS,EEEE,DD   S - Start Address
                               E - End Address
                               D - Data pattern in HEX

Terminal prints address of each location that contains DD within the S - E range.

SEARCH FOR A 16 BIT WORD IN MEMORY

        SW,SSSS,EEEE,DDDD   S - Start Address
                                 E - End Address
                                 D - Data pattern in HEX

Terminal prints address of each point in memory that contains two successive 8 bit bytes DDDD.

CALL A SUBROUTINE FROM MONITOR

        CS,DDDD   D - Address of Subroutine

Subroutine (or another program) executes, and if terminated with a 39$_{16}$ (RTS), Monitor reenters command mode.

CONVERT ASCII TO HEX EQUIVALENT

        CA,A   A - Any Valid ASCII
                        Character

Terminal prints HEX equivalent, and Monitor returns to command mode.

TEST MEMORY ROUTINE

        TM,SSSS,EEEE   S - Start Address
                            E - End Address

Routine responds by clearing locations S through E inclusive. If at any time the start address of the GP Monitor is reached, program aborts and Monitor command mode is reentered.

When memory is cleared, program sequentially increments memory and tests for valid result. This is done for all 256 combinations per bytes for all memory locations selected. An 8K RAM requires about two minutes to test.

4K · 8K · 16K or 32K Configuration

By William E. Warren

Any error in memory causes terminal to print address where error occurred.

When routine finishes, Monitor returns to command mode.

LOAD A HEX FORMATTED OBJECT TAPE
(Must Be Continuous Data)

LT,DDDD — D - Start Address of Contiguous Memory

Object tape is loaded into memory until S9 is read. Monitor returns to command mode.

PUNCH FOLD-MARK FORMATTED PAPER TAPE

PT,SSSS,EEEE — S - Start Address
E - End Address

Terminal with paper tape punch outputs object tape with 8½" rubout fold marks.

Leaders and trailers are written to tape with S9 terminator included.

PUNCH A BNPF FORMATTED OBJECT TAPE

PB,SSSS,EEEE — S - Start Address
E - End Address

Terminal punches paper tape suitable for most Intel MDS prom-programming system readers.

Leaders, foldmarks and trailers are written to tape.

CALCULATE HEX OFFSET

```
0010   20   HERE   BRA   THERE
       (30)        CO,0010,0042__30
                              ANSWER PRINTED
                              BY TERMINAL
       .
0042   20   THERE  BRA   HERE
       (CC)        CO,0042,0010__CC
                              ANSWER PRINTED
                              BY TERMINAL
```

When finished, Monitor returns to command mode. If branch is out of range, terminal prints an X, and Monitor reenters command mode.

MOVE A BLOCK OF CONTIGUOUS MEMORY

MM,SSSS,EEEE,DDDD — S - Source Start Address
E - Source End Address
D - Destination Start Address

Terminal prints destination end address when block move is complete and Monitor reenters command mode.

CLEAR CONTIGUOUS MEMORY BLOCK

CM,SSSS,EEEE — S - Start Address
E - End Address

Routine sets all bits in all memory locations to zero. If Monitor starting address is reached, program aborts and Monitor reenters command mode.

P1 Call user Program #1
User selected Address

P2 Call user Program #2
User selected Address

(ESC,ESC) Escape,Escape
Jump to alternate Monitor

GP MONITOR 4K VERSION

```
2TS00B00004D4F4E49544F5220AC
S11E0B3E8EA0428D32BD0C2ECE0BDFE1002604A101270E080808088C0C2726D3
S11E0B59EFBD0BBD20DFEE02AD0020D9CE0B6B7E0FCC0D0A00000FF000000C1
S11E0B74048DEECE0B7C20EC5245414459 3E048D19CE0B8C8DDE20B22A2A2A05
S11E0B8F41424F52542A2A2A048C0B3E27E639FFA04ABD0E91CE0BAC8DBF7E9E
S11E0BAA0CB20D0A000000455252 4F5220415420 2D2004CE0BC37E0FCC204949
S11E0BC54E56414C49442 0434F4D4D414E442004C6404FBD0FFA5A26F9394CF7
S11E0BE0440CF344440D214C4D0D26444D0DBB53420D68434D0D5543530C9D9D
S11E0BFB4D4D0CB8 53570D94 4C540E7050540ED5544D0E25434F0DE34341 0EA5
S11E0C16621B1BE0D050420F6450310B3E50320B3EFF7E0B837E0B3EBD0FFD42
S11E0C31167E0FFD80302B0F81092F0A81112B078 1162E038007398 1EB26D0C9D
S11E0C4C20DD8DE220E3378DF948 4848 48168DF21B36BBA05EB7A05E3233394B
S11E0C6737368DE7168DE4CEA04AE700A701EE00323339 8DB6811B27AB812CD5
S11E0C8226F6398DF320C96DEF20DA8DFAFFA04E8DF5FFA050 8DF0FFA0523993
S11E0C9D8DEAAD00 7E0B3E8DE3FFA0588DDEFFA05A39BD0B65CEA04A7E0FF6E1
S11E0CB88DD3FEA04EA600BCA0502719 08FFA04EFEA052BD0B98A700A100278B
S11E0CD3037E0B8308FFA05220DDFEA052A700BD0FF2CEA0527E0FF67E0C8952
S11E0CEE7E0C7A20B18DF67FA05FBD0FFD811B271AFEA04ABD0B98A700A100DB
S11E0D0927037E0B8 308FFA04A7CA05F270220DF8D94B6A05F7E0F468DC87E85
S11E0D248FCC8DC37FA05FBD0CAF8DBEBD0C527CA05FFEA04ABD0B98A700A11E
S11E0D3F0027037E0B8308FFA04AB6A05F811027D8BD0FF220DB8D9AFEA05853
S11E0D5ABD0B986F00BCA05A27030820F3398D67FEA058FFA04ABD0C8516FE22
S11E0D75A04AA600112708BCA05A27E60820F3FFA04A37BD0CAF33FEA04A20DE
S11E0D90F07E0C8D8DFBFEA04EBCA05027C9A600E601B1A0522605F1A05327C7
S11E0DAB0308 20EAFFA04A08FFA04EBD0CAF20DBBD0CA4FEA058FFA04ABD0CAE
S11E0DC6AF8610B7A05FFEA04A09BCA05A27920 5BD0F0FFA04A7AA05F26EB7C
S11E0DE120DEBD0CA45F5A5AFEA058B6A058B1A05A2215B6A059B1A05B220D65
S11E0DFCBCA05A2715085CC180271620F3BCA05A2708095AC17F270920F3BD69
S11E0E170FF2177E0F46BD0FF286587E0FFABD0CA4BD0D57FEA058A6004CA791
S11E0E3200B7A060E600F7A061112608BCA05A271A0820E8FFA04ABD0B9ECEA9
S11E0E4DA060BD0FF0BD0FF0FEA04AB6A06020D381FF26CA39BD0C7ABD0FFDC8
S11E0E6836BD0FF2327E0F46BD0C89FFA04E86118D58863CB78007BD0FFD816D
S11E0E835326F9BD0FFD81312711813926EE8634B780078613 8D38397E0C52F2
S11E0E9E7FA05E8DF88002B7A05DBD0C67FEA04E8DEBBD0B987AA05D2709A7BB
S11E0EB900A100260B0820EDFFA04E7CA05E27B6863F8D058DC27E0B97E0F95
S11E0ED4FABD0CA486128DF6BD0BD5FEA058FFA04CB6A05BB0A04DF6A05AF2CF
S11E0EEFA04AC260481202502861F8B04B7A05C8003B7A05DBD0B65CE0F61BDC0
S11E0F0A0FCC7FA05ECEA05C8D39CEA04C8D44FEA04C8D2F7AA05D26F9FFA07A
S11E0F254C73A05ECEA05E8D1FFEA04C09BCA05A26AE86538D976398D93BD62
S11E0F400BD58614208C3630BD0FE3323937BD0FE309E60008FBA05EF7A05E21
S11E0F5B33398DEE20EC533104BD0CA4FEA05A08FFA04E8612BD0FFABD0BD5A7
S11E0F768D2C2003BD0B65C6088 6428D77FEA058BCA05A2711A60008FFA05830
S11E0F918D1B86468D635A26E520DEC6648D068 61420 56C64D86FF8D505A26B8
S11E0FACF9 3937C608360C465A26FAC608324624068 6508D3920004 64E8D3329
S11E0FC75A26EF3339A600810427F98D260820F5840F8109230280B078D302066
S11E0FE217A600474747478DECA6000820E78DF186202004 8DEB20F67EE1D1E3
S1060FFD7EE1ACE2
S9030000FC
```

GP MONITOR 8K VERSION

```
2TS00B00004D4F4E49544F5220AC
S11E1B3E8EA0428D32BD1C2ECE1BDFE1002604A101270E080808088C1C272693
S11E1B59EFBD1BBD20DFEE02AD0020D9CE1B6B7E1FCC0D0A00000FF00000081
S11E1B74048DEECE1B7C20EC524541445 93E048D19CE1B8C8DDE20B22A2A2AD5
S11E1B8F41424F52542A2A2A048C1B3E27E639FFA04ABD1E91CE1BAC8DBF7E5E
S11E1BAA1CB20D0A000000455252 4F5220415420 2D2004CE1BC37E1FCC204999
S11E1BC54E56414C49 4420434F4D4D414E442004C6404FBD1FFA5A26F9394CD7
S11E1BE0441CF344441D214C4D1D26444D1DBB53421D68434D1D5543531C9D1D
S11E1BFB4D4D1CB853571D94 4C541E7050541ED5544D1E25434F1DE343411E25
S11E1C16621B1BE0D050421F6450311B3E50321B3EFF7E1B837E1B3EBD1FFD2
S11E1C31167E1FFD80302B0F81092F0A81112B078 1162E038007398 1EB26DC7D
S11E1C4C20DD8DE220E3378DF948484848 168DF21B36BBA05EB7A05E3233393B
S11E1C6737368DE7168DE4CEA04AE700A701EE0032333 98DB6811B27AB812CC5
S11E1C8226F6398DF320C96DEF20DA8DFAFFA04E8DF5FFA052BD1B98A700A100
S11E1C9D8DEAAD007E1B3E8DE3FFA0588DDEFFA05A39BD1B65CEA04A7E1FF6A1
S11E1CB88DD3FEA04EA600BCA050271908FFA04EFEA052BD1B98A700A100276B
S11E1CD3037E1B8308FFA05220DDFEA052A700BD1FF2CEA0527E1FF67E1C8902
S11E1CEE7E1C7A20B18DF67FA05FBD1FFD811B271AFEA04ABD1B98A700A1009B
S11E1D0927037E1B8308FFA04A7CA05F270220DF8D94B6A05F7E1F468DC87E55
S11E1D241FCC8DC37FA05FBD1CAF8DBEBD1C527CA05FFEA04ABD1B98A700A1CE
S11E1D3F0027037E1B8308FFA04AB6A05F811027D8BD1FF220DB8D9AFEA05D83
S11E1D5ABD1B986F00BCA05A27030820F3398D67FEA058FFA04ABD1C8516FEF2
S11E1D75A04AA600112708BCA05A27E60820F3FFA04A37BD1CAF33FEA04A20BE
S11E1D90F07E1C8D8DFBFEA04EBCA05027C9A600E601B1A0522605F1A05327A7
S11E1DAB0308 20EAFFA04A08FFA04EBD1CAF20DBBD1CA4FEA058FFA04ABD1C6E
S11E1DC6AF8610B7A05FFEA04A09BCA05A279208BD1FF0FFA04A7AA05F26EB5C
S11E1DE120DEBD1CA45F5A5AFEA058B6A058B1A05A2215B6A059B1A05B220D45
S11E1DFCBCA05A2715085CC180271620F3BCA05A2708095AC17F270920F3BD59
S11E1F171FF2177E1F46BD1FF286587E1FFABD1CA4BD1D57FEA058A6004CA721
S11E1E3200B7A060E600F7A061112608BCA05A271A0820E8FFA04ABD1B9ECE89
S11E1E4DA060BD1FF0BD1FF0FEA04AB6A06020D381FF26CA39BD1C7ABD1FFD78
S11E1E6836BD1FF2327E1F46BD1C89FFA04E86118D58863CB78007BD1FFD811D
S11E1E835326F9BD1FFD813127118 13926EE8634B780078 6138D38397E1C52C2
S11E1E9E7FA05E8DF88002B7A05DBD1C67FEA04E8DEBBD1B987AA05D2709A78B
S11E1E900A100260B0820EDFFA04E7CA05E27B6863F8D058DC27E1B9E7E1F65
S11E1ED4FABD1CA486128DF6BD1BD5FEA058FFA04CB6A05BB0A04DF6A05AF29F
S11E1EEFA04AC260481202502861F8B04B7A05C8003B7A05DBD1B65CE1F61BD90
S11E1F0A1FCC7FA05ECEA05C8D39CEA04C8D44FEA04C8D2F7AA05D26F9FFA05A
S11E1F254C73A05ECEA05E8D1FFEA04C09BCA05A26AE86538D976398D93BD52
S11E1F401BD58614208C3630BD1FE3323937BD1FE309E60008FBA05EF7A05EE1
S11E1F5B33398DEE20EC533104BD1CA4FEA05A08FFA04E8612BD1FFABD1BD567
S11E1F768D2C2003BD1B65C60886428D77FEA058BCA05A2711A60008FFA05810
S11E1F918D1B86468D635A26E520DEC6648D068 61420 56C64D86FF8D505A26A8
S11E1FACF9 3937C608360C465A26FAC608324624068 6508D3920004 64E8D3319
S11E1FC75A26EF3339A600810427F98D260820F5840F8109230280B078D302056
S11E1FE217A600474747478DECA6000820E78DF186202004 8DEB20F67EE1D1D3
S1061FFD7EE1ACD2
S9030000FC
```

GP MONITOR 16K VERSION

```
2TS00B00004D4F4E49544F5220AC
S11E3B3E8EA0428D32BD3C2ECE3BDFE1002604A101270E080808088C3C272613
S11E3B59EFBD3BBD20DFEE02AD0020D9CE3B6B7E3FCC0D0A000000FF0000001
S11E3B7404BDEECE3B7C20EC5245414459E048D19CE3B8C8DDE20B22A2A2A75
S11E3B8F41424F52542A2A2A048C3B3E27E639FFA04ABD3E91CE3BAC8DBF7EDE
S11E3BAA3CB20D0A000000455252F52204154202D2004CE3BC37E3FCC204989
S11E3BC54E56414C494420434F4D4D414E442004C6404FBD7FFA5A26F9394C97
S11E3BE0443CF344443D214C4D3D26444D3DBB53423D68434D3D5543533C9D1D
S11E3BFB4D4D3CB8535730D944C543E7050543ED5544D3E25434F3DE343413E25
S11E3C16621B1BE0D050423F6450313B3E50323B3EFF7E3B837E3B3E8D7FFD2
S11E3C3116E7E3FFD80302B0F81092F0A81112B0781162E038007398 1EB26DC3D
S11E3C4C20DD8DE220E3378DF948484848168DF21B36BBA05EB7A05E3233391B
S11E3C6737368DE7168DE4CEA04AE700A701EE003233398DB6811B27AB812CA5
S11E3C8226F6398DF320C98DEF20DA8DFAFFA04E8DF5FFA0508DF0FFA0523963
S11E3C9D8EAAD007E3B3E8DE3FFA0588DDEFFA05A39BD3B65CEA04A7E3FF621
S11E3CB88DD3FEA04EA600BCA050271908FFA04EFEA052BD7B98A700A100272B
S11E3CD3037E3B8308FFA05220DDFEA052A700BD3FF2CEA0527E3FF67E3C8962
S11E3CEE7E3C7A20B18DF67FA05FBD3FFD811B271AFEA04ABD3B98A700A1001B
S11E3D0927037E3B8308FFA04A7CA05F270220DF8D94B6A05F7E3F468DC87EF5
S11E3D243FCC8DC37FA05FBD3CAF8DBEBD3C527CA05FFEA04ABD3B98A700A12E
S11E3D3F0027037E3B8308FFA04AB6A05F811027D8BD3FF220DB8D9AFEA058C3
S11E3D5ABD3B986F00BCA05A270308 20F3398D87FEA058FFA04ABD3C8516FE92
S11E3D75A04AA6001127 08BCA05A27E608 20F3FFA04A37BD3CAF33FEA04A207E
S11E3D90F07E3C8D8DFBFEA04EBCA05027C9A600E601B1A0522605F1A0532767
S11E3DAB0308 20EAFFA04A08FFA04EBD3CAF20DBBD3CA4FEA058FFA04ABD3CEA
S11E3DC6A4F8610B7A05FFEA04A09BCA05A279208BD3FF0FFA04A7AA05F26EB1C
S11E3DE120DEBD3CA45F5A5AFEA058B6A05B81A05A2215B6A059B1A05B220D05
S11E3DFC8CA05A2715085CC180271620F3BCA05A2708095AC17F270920F3BD39
S11E3E173FF2177E3F46BD3FF286587E3FFABD3CA4BD3D57FEA058A6004CA741
S11E3E3200B7A060E600F7A061112609BCA05A271A0820E8FFA04ABD3B9ECE49
S11E3E4DA060BD3FF0BD3FF0FEA04AB6A60620D381FF26CA39BD3C7ABD3FFDD8
S11E3E6836BD3FF2327E3F46BD7C89FFA04E86118D58863CB78007BD3BD5E7
S11E3E835326F9BD3FFD8131271181 3926EE8634B7800786138D38397E3C5262
S11E3E9E7FA05E8DF88002B7A05DBD7C67FEA04E8DEBBD7B987AA05D2709A78
S11E3EB900A100260B08 20EDFFA04E7CA05E27B6863F8D058DC27E3B9E7E7F05
S11E3ED4FABD3CA486128DF6BD3BD5FEA058FFA04CB6A05BB0A04DF6A05AF23F
S11E3EEFA04C26048120250286 1F8B04BA7A05C8D78007BD3FFF621
S11E3F0A3FCC7FA05ECEA05C8D39CEA04C8D44FEA04C8D2F7AA05D26F9FFA01A
S11E3F254C73A05ECEA05E8D1FFEA04C09BCA05A26AE8 6538D978 6398D93BD32
S11E3F403BD58614208C3630BD3FE3323937BD3FE309E60008FBA05EF7A05E61
S11E3F5B33398DEE20EC5331 04BD3CA4FEA05A08FFA04E8612BD3FFABD3BD5E7
S11E3F768D2C2003BD3B65C60886428D77FEA058BCA05A2711A60008FFA058D0
S11E3F918D1B86468D635A26E520DEC6648D068 61420 56C64D8 6FF8D505A2648
S11E3FACF9397C608 360C465A26FAC608 32462408 6508D3920048 64E8D33F9
S11E3FC75A26EF3339A600810427F98D260820F5840F810923028B078B302036
S11E3FE217A600474747478DECA6000820E78DF1862020048DEB20F67EE1D1B3
S1063FFD7EE1ACB2
S9030000FC
```

GP MONITOR 32K VERSION

```
2TS00B00004D4F4E49544F5220AC
S11E7B3E8EA0428D32BD7C2ECE7BDFE1002604A101270E080808088C7C272613
S11E7B59EFBD7BBD20DFEE02AD0020D9CE7B6B7E7FCC0D0A000000FF0000001
S11E7B7404BDEECE7B7C20EC5245414459E048D19CE7B8C8DDE20B22A2A2A85
S11E7B8F41424F52542A2A2A048C7B3E27E639FFA0ABD7E91CE7BAC8DBF7EDE
S11E7BAA7CB20D0A000000455252F52204154202D2004CE7BC37E3FCC204989
S11E7BC54E56414C494420434F4D4D414E442004C6404FBD7FFA5A26F9394C17
S11E7BE0447CF344443D214C4D3D26444D3DBB53427D68434D7D5543537C9D1D
S11E7BFB4D4D7CB8535730D944C547E7050547ED5544D7E25434F7DE343417E25
S11E7C16621B1BE0D050423F6450317B3E50327B3EFF7E7B837E7B3E8D7FFD32
S11E7C3116E7E7FFD80302B0F81092F0A81112B0781162E038007398 1EB26DCBD
S11E7C4C20DD8DE220E3378DF948484848168DF21B36BBA05EB7A05E323339DB
S11E7C6737368DE7168DE4CEA04AE700A701EE003233398DB6811B27AB812C65
S11E7C8226F6398DF320C98DEF20DA8DFAFFA04E8DF5FFA0508DF0FFA0523923
S11E7C9D8EAAD007E7B3E8DE3FFA0588DDEFFA05A39BD7B65CEA04A7E7FF621
S11E7CB88DD3FEA04EA600BCA050271908FFA04EFEA052BD7B98A700A1002 7AB
S11E7CD3037E7B8308FFA0522 0DDFEA052A700BD7FF2CEA0527E7FF67E7C8922
S11E7CEE7E7C7A20B18DF67FA05FBD7FFD811B271AFEA04ABD7B98A700A1001B
S11E7D0927037E7B8308FFA04A7CA05F270220DF8D94B6A05F7E7F468DC87E35
S11E7D247FCC8DC37FA05FBD7CAF8DBEBD7C527CA05FFEA04ABD7B98A700A1EE
S11E7D2F0027037E7B8308FFA04AB6A05F811027D8BD7FF220DB8D9AFEA05803
S11E7D5ABD7B986F00BCA05A270308 20F3398D87FEA058FFA04ABD7C8516FED2
S11E7D75A04AA6001127 08BCA05A27E608 20F3FFA04A37BD7CAF33FEA04A20FE
S11E7D90F07E7C8D8DFBFEA04EBCA05027C9A600E601B1A0522605F1A0532F7E
S11E7DAB0308 20EAFFA04A08FFA04EBD7CAF20DBBD7CA4FEA058FFA04ABD7CEE
S11E7DC6AF8610B7A05FFEA04A09BCA05A279208 BD7FF0FFA04A7AA05F26EB9C
S11E7DE120DEBD7CA45F5A5AFEA058B6A05B81A05A2215B6A059B1A05B220D85
S11E7DFC8CA05A2715085CC180271620F3BCA05A2708095AC17F270920F3BDBC
S11E7E177FF2177E7F46BD7FF286587E7FFABD7CA4BD7D57FEA058A6004CA781
S11E7E3200B7A060E600F7A061112608BCA05A271A0820E8FFA04ABD7B9ECEC9
S11E7E4DA060BD7FF0BD7FF0FEA04AB6A60620D381FF26CA39BD7C7ABD7FFD98
S11E7E6836BD7FF2327E7F46BD7C89FFA04E86118D58863CB78007BD7FFDB13D
S11E7E835326F9BD7FFD8131271181 3926EE8634B7800786138D38397E7C52A2
S11E7E9E7FA05E8DF88002B7A05DBD7C67FEA04E8DEBBD7B987AA05D2709A76B
S11E7EB900A100260B08 20EDFFA04E7CA05E27B6863F8D058DC27E7B9E7E7F45
S11E7ED4FABD7CA486128DF6BD7BD5FEA058FFA04CB6A05BB0A04DF6A05AF27F
S11E7EEFA04C26048120250286 1F8B04BA7A05C8D78007BD7FFF621
S11E7F0A7FCC7FA05ECEA05C8D39CEA04C8D44FEA04C8D2F7AA05D26F9FFA09A
S11E7F254C73A05ECEA05E8D1FFEA04C09BCA05A26AE8 6538D978 6398D93BDF2
S11E7F407BD58614208C3630BD7FE3323937BD7FE309E60008FBA05EF7A05E61
S11E7F5B33398DEE20EC5331 04BD7CA4FEA05A08FFA04E8612BD7FFABD7BD5E7
S11E7F768D2C2003BD7B65C60886428D77FEA058BCA05A2711A60008FFA05850
S11E7F918D1B86468D635A26E520DEC6648D068 61420 56C64D8 6FF8D505A2648
S11E7FACF9397C608 360C465A26FAC608 324624 08 6508D3920048 64E8D33B9
S11E7FC75A26EF3339A600810427F98D260820F5840F810923028B078B3020F6
S11E7FE217A600474747478DECA6000820E78DF1862020048DEB20F67EE1D173
S1067FFD7EE1AC72
S9030000FC
```

SAMPLE RUN

```
*M A048
*A048  9C   0B
*A049  FD   3E
*A04A  0B
*R 7E AE FF 9DFF 0B3E A042
*G
READY>LD,0000 THIS IS A TEST .
0013 13
READY>DD,0000 THIS IS A TEST .
READY>LM,0000
0000 ,00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
0010 ,00 11 22 33 M  ◄─ NON-HEX CHARACTER
ERROR AT - 0014 ***ABORT***
READY>DM,0000,0014
0000 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
0010 00 11 22 33 00
READY>LM,0722
0722 ,12 34
READY>SB,0000,07FF,12   ⎫  SEARCH
0722                    ⎬  ROUTINES
READY>SW,0000,07FF,1234 ⎭
0722
READY>DM,0B3E,0B4F
0B3E 8E A0 42 8D 32 BD 0C 2E CE 0B DF E1 00 26 04 A1
0B4E 01 27
READY>MM,0B3E,0B4F,0000 0011  NEW ENDING ADDRESS
READY>DM,0000,0011
0000 8E A0 42 8D 32 BD 0C 2E CE 0B DF E1 00 26 04 A1
0010 01 27
READY>CO,0000,0045 43 ⎫
READY>CO,0045,0000 B9 ⎬  OFFSET CALCULATION
READY>CA,G 47          ⎭
READY>CA,K 4B
READY>CA,L 4C
READY>PB,0000,0003

        BPNNNN PPPN FBPN PNNNNN FBN PNNNNN PN FB

READY>TM,0000,07FF
READY>TM,0000,0FFF
ERROR AT - 0B3E ***ABORT***  ◄ ATTEMPT TO DAMAGE MONITOR
READY>PT,0000,003F
S12300000000000000000000000000000000000000000000000000000000000000000000DC

S1230020000000000000000000000000000000000000000000000000000000000000000BC
9                                       S
READY>TM,0000,07FF
READY>DM,0100,010F
0100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0110
READY>LT,0100READY>HH INVALID COMMAND
READY>DM,0100,010F
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110
READY>CS,E1D1,
READY>P1
READY>P2
READY>
*G
READY> INVALID COMMAND
READY>
```

PROGRAM LISTING

```
00001                   NAM   MONITOR
00002                   OPT   O,NOG
00003         •
00004         *M6800 GP MONITOR VER 2.1
00005         *REVISED JAN 7/1978
00006         *WRITTEN BY  WILLIAM E. WARREN
00007         •
00008         •
00009         •
00010         •
00011         *THIS VERSION FOR 4K RAM SYSTEMS
00012         •
00013         *EQUATES AND REGISTERS
00014         •
00015  A042   STACK  EQU  $A042   DEFINE PROGRAM STACK
00016  8007   RDRCON EQU  $8007   READER CONTROL REGISTER
00017  E0D0   ALTMON EQU  $E0D0   ALTERNATE MONITOR
00018  A04A          ORG  STACK+8
00019  A04A   BF4HEX RMB  2       GENERAL
00020  A04C 0002 TEMPX  RMB  2
00021  A04E 0002 TEMPX1 RMB  2     INDEX
00022  A050 0002 TEMPX2 RMB  2
00023  A052 0002 TEMPX3 RMB  2     REGISTER
00024  A054 0002 TEMPX4 RMB  2
00025  A056 0002 TEMPX5 RMB  2     STORAGE
00026  A058 0002 STARTX RMB  2     START ADDRESS VECTOR
00027  A05A 0002 ENDX   RMB  2     END ADDRESS VECTOR
00028  A05C 0001 FRMCNT RMB  1     FRAMECOUNT REGISTER
00029  A05D 0001 BYTCNT RMB  1     BYTE COUNT REGISTER
00030  A05E 0001 CHKSUM RMB  1     CHECKSUM REGISTER
00031  A05F 0001 COUNTU RMB  1     8 BIT COUNTER
00032  A060 0002 BUFFER RMB  2     SPECIAL STORAGE
00033         •
```

```
00034                  *THIS MONITOR COMMAND ROUTINE
00035                  *ACCEPTS TWO-INPUT COMMAND MNEMONICS
00036                  *THAT CALL UP THE REQUIRED SUBROUTINE
00037                  *
00038                  *IF AN ERROR IS MADE ON ENTRY
00039                  *THE TERMINAL PRINTS 'INVALID COMMAND'
00040                  *AND THE MONITOR IS REENTERED .
00041                  *
00042 0B3E             ORG      $0FFF-$4C1
00043                  *
00044 0B3E 8E A042 MONITR LDS    #STACK    SET STACK
00045 0B41 8D 32             BSR   PREADY    DO PROMPT
00046 0B43 BD 0C2E           JSR   IN2ASC    INPUT COMMANDS
00047 0B46 CE 0BDF           LDX   #DATA6    POINT TO TABLE
00048 0B49 E1 00    FINDC    CMP B 0,X       1ST CHAR?
00049 0B4B 26 04             BNE   NEXT4     NOPE
00050 0B4D A1 01             CMP A 1,X       2ND CHAR?
00051 0B4F 27 0E             BEQ   FOUNDC    YES
00052 0B51 08      NEXT4     INX             INCREMENT
00053 0B52 08                INX             TO
00054 0B53 08                INX             NEXT
00055 0B54 08                INX             COMMAND
00056 0B55 8C 0C27           CPX   #DATA6E   END YET?
00057 0B58 26 EF             BNE   FINDC     KEEP LOOKING
00058 0B5A BD 0BBD           JSR   INVALD    INVALID COMMAND
00059 0B5D 20 DF             BRA   MONITR    BACK TO START
00060 0B5F EE 02   FOUNDC    LDX   2,X       FETCH ROUTINE ADDRESS
00061 0B61 AD 00             JSR   0,X       DO THE ROUTINE
00062 0B63 20 D9             BRA   MONITR    GO BACK TO START
00063                  *
00064                  *START A NEW LINE
00065                  *
00066 0B65 CE 0B6B NEWLIN    LDX   #NEWLDT   POINT TO DATA
00067 0B68 7E 0FCC PDATA1    JMP   PDATA     PRINT STRING
00068 0B6B 0D      NEWLDT    FCB   $D,$A,0,0,0,$FF,0,0,0,4
00069                  *
00070                  * READY> PRINTOUT SUBROUTINE
00071                  *
00072 0B75 8D EE   PREADY    BSR   NEWLIN    START NEW LINE FIRST
00073 0B77 CE 0B7C           LDX   #DATA2
00074 0B7A 20 EC             BRA   PDATA1    PRINT STRING
00075 0B7C 52      DATA2     FCC   'READY>'
00076 0B82 04                FCB   $4
00077                  *
00078                  *ABORT PRINTOUT SUBROUTINE
00079                  *
00080 0B83 8D 19   PABORT    BSR   PERROR
00081 0B85 CE 0B8C           LDX   #DATA3
00082 0B88 8D DE             BSR   PDATA1
00083 0B8A 20 B2             BRA   MONITR
00084 0B8C 2A      DATA3     FCC   '***ABORT***'
00085 0B97 04                FCB   $4
00086                  *
00087                  *MONITOR OVERWRITE PROTECTION ROUTINE
00088                  *
00089 0B98 8C 0B3E DANGER    CPX   #MONITR   IS X NEAR MONITOR?
00090 0B9B 27 E6             BEQ   PABORT    YES GET OUT QUICK
00091 0B9D 39                RTS             NO ITS OK
00092                  *
00093                  *ERROR PRINTOUT SUBROUTINE
00094                  *
00095 0B9E FF A04A PERROR    STX   BF4HEX    SAVE ADDRESS OF ERROR
00096 0BA1 BD 0E91           JSR   OUT       TURN OFF READER(IF ON)
00097 0BA4 CE 0BAC           LDX   #DATA4    POINT AT MESSAGE
00098 0BA7 8D BF             BSR   PDATA1    PRINT IT
00099 0BA9 7E 0CB2           JMP   LADDR     PRINT ADDRESS
00100 0BAC 0D      DATA4     FCB   $D,$A,0,0,0
00101 0BB1 45                FCC   'ERROR AT -
00102 0BBC 04                FCB   4
00103                  *
00104                  *INVALID COMMAND MESSAGE
00105                  *
00106 0BBD CE 0BC3 INVALD    LDX   #IVAPRT
00107 0BC0 7E 0FCC           JMP   PDATA
00108 0BC3 20      IVAPRT    FCC   ' INVALID COMMAND'
00109 0BD4 04                FCB   $4
00110                  *
00111                  *PUNCH 64 NULL LEADER/TRAILER
00112                  *
00113 0BD5 C6 40   P64NUL    LDA B #64
00114 0BD7 4F      PNUL      CLR A
00115 0BD8 BD 0FFA           JSR   CPRINT
00116 0BDB 5A                DEC B
00117 0BDC 26 F9             BNE   PNUL
00118 0BDE 39                RTS
00119                  *
00120                  *COMMAND TABLE
00121                  *
00122 0BDF 4C      DATA6     FCC   'LD' LOAD DATA
00123 0BE1 0CF3              FDB   LODAT
00124 0BE3 44                FCC   'DD' DUMP DATA
00125 0BE5 0D21              FDB   DUDAT
00126 0BE7 4C                FCC   'LM' LOAD MEMORY
00127 0BE9 0D26              FDB   LOMEM
00128 0BEB 44                FCC   'DM' DUMP MEMORY
00129 0BED 0DBB              FDB   DUMEM
00130 0BEF 53                FCC   'SB' SEARCH MEMORY FOR 8 BIT BYTE
00131 0BF1 0D68              FDB   SERMEM
00132 0BF3 43                FCC   'CM' CLEAR MEMORY
00133 0BF5 0D55              FDB   CLRMEM
00134 0BF7 43                FCC   'CS' CALL SUBROUTINE
00135 0BF9 0C9D              FDB   CALSUB
00136 0BFB 4D                FCC   'MM' MOVE MEMORY BLOCKS
00137 0BFD 0CB8              FDB   MOVMEM
00138 0BFF 53                FCC   'SW' SEARCH MEMORY FOR 16 BIT WORD
00139 0C01 0094              FDB   SERADD
00140 0C03 4C                FCC   'LT' LOAD TAPE WITH OFFSET
00141 0C05 0E70              FDB   OFLOAD
00142 0C07 50                FCC   'PT' PUNCH FORMATTED PAPER TAPE
00143 0C09 0ED5              FDB   PUNTAB
00144 0C0B 54                FCC   'TM' TEST MEMORY
00145 0C0D 0E25              FDB   TSTMEM
00146 0C0F 43                FCC   'CO' CALCULATE HEX OFFSET
00147 0C11 0DE3              FDB   CALOFF
00148 0C13 43                FCC   'CA' CONVERT ASCII TO HEX
00149 0C15 0E62              FDB   CONASC
00150 0C17 1B                FCB   $1B,$1B  (ESC,ESC) GO TO ALTERNATE MON-
00151 0C19 E0D0              FDB   ALTMON
00152 0C1B 50                FCC   'PB' PUNCH BNPF TAPE
00153 0C1D 0F64              FDB   PUNBNF
00154 0C1F 50                FCC   'P1' GO TO PROGRAM ONE
00155 0C21 0B3E              FDB   MONITR   USER VECTOR GOES HERE
00156 0C23 50                FCC   'P2' GO TO PROGRAM TWO
00157 0C25 0B3E              FDB   MONITR   USER VECTOR GOES HERE
00158 0C27 FF      DATA6E    FCB   $FF      END OF COMMAND TABLE
```

```
00159                  *
00160                  *CONNECTING JUMPS
00161                  *
00162 0C28 7E 0B83 JUMP1     JMP   PABORT
00163 0C2B 7E 0B3E JUMP2     JMP   MONITR
00164                  *
00165                  *INPUT 2 ASCII CHARACTERS INTO B AND A
00166                  *
00167 0C2E BD 0FFD IN2ASC    JSR   CINPUT    FIRST CHAR
00168 0C31 16                TAB             PUT IN B
00169 0C32 7E 0FFD INASC     JMP   CINPUT    FETCH SECOND CHAR AND RTS
00170                  *
00171                  *MAKE HEX FROM DATA IN A
00172                  *IF NON HEX THEN DO ERROR
00173                  *AND VECTOR BACK TO MONITOR
00174                  *
00175 0C35 80 30   MAKHEX    SUB A #$30     STRIP ASCII
00176 0C37 2B 0F             BMI   NOTHEX
00177 0C39 81 09             CMP A #$09      0 TO 9 HEX?
00178 0C3B 2F 0A             BLE   HEX
00179 0C3D 81 11             CMP A #$11
00180 0C3F 2B 07             BMI   NOTHEX
00181 0C41 81 16             CMP A #$16
00182 0C43 2E 03             BGT   NOTHEX
00183 0C45 80 07             SUB A #$07
00184 0C47 39      HEX       RTS             OK, EXIT
00185 0C48 81 EB   NOTHEX    CMP A #$EB     'ESC' KEY?
00186 0C4A 26 DC             BNE   JUMP1     HEX ERROR
00187 0C4C 20 DD             BRA   JUMP2     ESCAPE SELECTED
00188                  *
00189                  *INPUT ONE HEX INTO A
00190                  *
00191 0C4E 8D E2   IN1HEX    BSR   INASC     FETCH CHAR
00192 0C50 20 E3             BRA   MAKHEX
00193                  *
00194                  *INPUT 2 HEX INTO A
00195                  *UPDATE CHECKSUM
00196                  *
00197 0C52 37      IN2HEX    PSH B           SAVE B
00198 0C53 8D F9             BSR   IN1HEX
00199 0C55 48                ASL A
00200 0C56 48                ASL A
00201 0C57 48                ASL A
00202 0C58 48                ASL A
00203 0C59 16                TAB             SHIFT TO UPPER B
00204 0C5A 8D F2             BSR   IN1HEX    GET LOWER
00205 0C5C 1B                ABA             MAKE A BYTE
00206 0C5D 36                PSH A           SAVE DATA
00207 0C5E BB A05E           ADD A CHKSUM    FETCH CHECKSUM
00208 0C61 B7 A05E           STA A CHKSUM    UPDATE
00209 0C64 32                PUL A           RESTORE DATA
00210 0C65 33                PUL B           RESTORE B
00211 0C66 39                RTS             DONE
00212                  *
00213                  *INPUT 4 HEX INTO X AND
00214                  *ALSO STORE AT BF4HEX
00215                  *
00216 0C67 37      IN4HEX    PSH B
00217 0C68 36                PSH A           SAVE ACC
00218 0C69 8D E7             BSR   IN2HEX    FETCH HI BYTE
00219 0C6B 16                TAB             PUT IN B
00220 0C6C 8D E4             BSR   IN2HEX    FETCH LO BYTE
00221 0C6E CE A04A           LDX   #BF4HEX   POINT AT DATA
00222 0C71 E7 00             STA B 0,X
00223 0C73 A7 01             STA A 1,X       PUT IN BUFFER
00224 0C75 EE 00             LDX   0,X       FETCH INTO X
00225 0C77 32                PUL A
00226 0C78 33                PUL B
00227 0C79 39                RTS
00228                  *
00229                  *CONTINUE (,) OR ESCAPE (ESC)
00230                  *
00231 0C7A 8D B6   CONTIN    BSR   INASC     FETCH CHAR
00232 0C7C 81 1B             CMP A #$1B     'ESC ?'
00233 0C7E 27 AB             BEQ   JUMP2
00234 0C80 81 2C             CMP A #',      COMMA ?
00235 0C82 26 F6             BNE   CONTIN    NO JUST WAIT
00236 0C84 39                RTS             DONE , GO
00237                  *
00238                  *CONTROLLED INPUT 2 HEX
00239                  *IN FORM ,HH
00240                  *
00241 0C85 8D F3   CIN2HX    BSR   CONTIN
00242 0C87 20 C9             BRA   IN2HEX    FETCH BYTE
00243                  *
00244                  *CONTROLLED INPUT 4 HEX
00245                  *IN FORM ,HHHH
00246                  *
00247 0C89 8D EF   CIN4HX    BSR   CONTIN    WAIT FOR COMMA
00248 0C8B 20 DA             BRA   IN4HEX
00249                  *
00250                  *INPUT 3 SETS OF 4 HEX
00251                  *IN FORM ,HHHH,HHHH,HHHH
00252                  *
00253 0C8D 8D FA   I3HEX4    BSR   CIN4HX
00254 0C8F FF A04E           STX   TEMPX1
00255 0C92 8D F5             BSR   CIN4HX
00256 0C94 FF A050           STX   TEMPX2
00257 0C97 8D F0             BSR   CIN4HX
00258 0C99 FF A052           STX   TEMPX3
00259 0C9C 39                RTS
00260                  *
00261                  *CALL SUBROUTINE POINTED TO
00262                  *BY ADDRESS IN X
00263                  *
00264 0C9D 8D EA   CALSUB    BSR   CIN4HX    FETCH THE ADDRESS
00265 0C9F AD 00             JSR   0,X       JUMP TO IT
00266 0CA1 7E 0B3E           JMP   MONITR    BACK TO MONITOR
00267                  *
00268                  *INPUT TWO SETS OF 4 HEX
00269                  *IN FORM ,HHHH,HHHH
00270                  *
00271 0CA4 8D E3   IN2HX4    BSR   CIN4HX
00272 0CA6 FF A058           STX   STARTX    FIRST ADDRESS
00273 0CA9 8D DE             BSR   CIN4HX
00274 0CAB FF A05A           STX   ENDX      SECOND ADDRESS
00275 0CAE 39                RTS
00276                  *
00277                  *START NEW LINE AND PRINT ADDRESS
00278                  *
00279 0CAF BD 0B65 NLADDR    JSR   NEWLIN
00280 0CB2 CE A04A LADDR     LDX   #BF4HEX   POINT AT DATA
00281 0CB5 7E 0FF6           JMP   P4HEXS    PRINT IT
00282                  *
00283                  *MOVE MEMORY BLOCK ROUTINE
```

```
00284                         *
00285  0CB8 8D D3  MOVMEM BSR   I3HEX4    FETCH PARAMETERS
00286  0CBA FE A04E DRCTMM LDX   TEMPX1    DIRECT ENTRY POINT
00287  0CBD A6 00          LDA A  0,X      SOURCE DATA
00288  0CBF BC A050          CPX   TEMPX2
00289  0CC2 27 19            BEQ   LASTBY
00290  0CC4 08              INX
00291  0CC5 FF A04E          STX   TEMPX1
00292  0CC8 FE A052          LDX   TEMPX3    DESTINATION
00293  0CCB BD 0B98          JSR   DANGER    BE CAREFUL
00294  0CCE A7 00            STA A  0,X      STORE DATA
00295  0CD0 A1 00            CMP A  0,X      CHECK IF THERE
00296  0CD2 27 03            BEQ   DRC1      ITS OK
00297  0CD4 7E 0B83          JMP   PABORT    NO MEMORY
00298  0CD7 08        DRC1   INX
00299  0CD8 FF A052          STX   TEMPX3
00300  0CDB 20 DD            BRA   DRCTMM
00301  0CDD FE A052 LASTBY LDX   TEMPX3
00302  0CE0 A7 00            STA A  0,X
00303  0CE2 BD 0FF2          JSR   SPACE
00304  0CE5 CE A052          LDX   #TEMPX3   POINT TO DATA
00305  0CE8 7E 0FF6          JMP   P4HEXS    PRINT IT AND EXIT
00306                        *
00307                  *LINKS
00308                        *
00309  0CEB 7E 0C89 JUMP3  JMP   CIN4HX
00310  0CEE 7E 0C7A JUMP4  JMP   CONTIN
00311  0CF1 20 B1   JUMP6  BRA   IN2HX4
00312                        *
00313                  *LOAD DATA INTO MEMORY
00314                        *
00315  0CF3 8D F6   LODAT  BSR   JUMP3     FETCH DESTINATION X
00316  0CF5 7F A05F          CLR   COUNTU
00317  0CF8 BD 0FFD LODAT1 JSR   CINPUT
00318  0CFB 81 1B            CMP A  #$1B     "ESC"?
00319  0CFD 27 1A            BEQ   QUITDA
00320  0CFF FE A04A          LDX   BF4HEX    FETCH ADDR
00321  0D02 BD 0B98          JSR   DANGER    BE CAREFUL
00322  0D05 A7 00            STA A  0,X
00323  0D07 A1 00            CMP A  0,X
00324  0D09 27 03            BEQ   LODAT2
00325  0D0B 7E 0B83          JMP   PABORT    NO RAM
00326  0D0E 08     LODAT2 INX
00327  0D0F FF A04A          STX   BF4HEX    RESTORE BUF
00328  0D12 7C A05F          INC   COUNTU    INCREMENT COUNT
00329  0D15 27 02            BEQ   QUITDA    YES EXIT
00330  0D17 20 DF            BRA   LODAT1    DO AGAIN
00331  0D19 8D 94   QUITDA BSR   NLADDR    PRINT ADDR
00332  0D1B B6 A05F          LDA A  COUNTU    FETCH BYTE COUNT
00333  0D1E 7E 0F46          JMP   P2HEXA    PRINT IT AND RTS
00334                        *
00335                  *DUMP DATA ROUTINE
00336                        *
00337  0D21 8D C6   DUDAT  BSR   JUMP3
00338  0D23 7E 0FCC          JMP   PDATA     PRINT STRING UNTIL EOT
00339                        *
00340             *LOAD MEMORY SEQUENTIALLY
00341             *WITH HEX DATA
00342                        *
00343  0D26 8D C3   LOMEM  BSR   JUMP3
00344  0D28 7F A05F LOMEM1 CLR   COUNTU
00345  0D2B BD 0CAF          JSR   NLADDR
00346  0D2E 8D BE            BSR   JUMP4     CONTIN
00347  0D30 BD 0C52 LOMEM2 JSR   IN2HEX
00348  0D33 7C A05F          INC   COUNTU    INCREMENT BYTE COUNT
00349  0D36 FE A04A          LDX   BF4HEX
00350  0D39 BD 0B98          JSR   DANGER    BE CAREFUL
00351  0D3C A7 00            STA A  0,X
00352  0D3E A1 00            CMP A  0,X      IS IT THERE?
00353  0D40 27 03            BEQ   LOMEM3    YES OK
00354  0D42 7E 0B83          JMP   PABORT    NOT THERE
00355  0D45 08     LOMEM3 INX
00356  0D46 FF A04A          STX   BF4HEX
00357  0D49 B6 A05F          LDA A  COUNTU
00358  0D4C 81 10            CMP A  #16
00359  0D4E 27 D8            BEQ   LOMEM1
00360  0D50 BD 0FF2          JSR   SPACE
00361  0D53 20 DB            BRA   LOMEM2
00362                        *
00363                  *CLEAR MEMORY ROUTINE
00364                        *
00365  0D55 8D 9A   CLRMEM BSR   JUMP6
00366  0D57 FE A058          LDX   STARTX
00367  0D5A BD 0B98 CLRM   JSR   DANGER    BE CAREFUL
00368  0D5D 6F 00            CLR   0,X       CLEAR A LOCATION
00369  0D5F BC A05A          CPX   ENDX      END YET?
00370  0D62 27 03            BEQ   DONE      YES EXIT
00371  0D64 08              INX
00372  0D65 20 F3            BRA   CLRM
00373  0D67 39      DONE   RTS
00374                        *
00375             *SEARCH MEMORY FOR 8 BIT BYTE
00376                        *
00377  0D68 8D 87   SERMEM BSR   JUMP6
00378  0D6A FE A058          LDX   STARTX
00379  0D6D FF A04A          STX   BF4HEX
00380  0D70 BD 0C85          JSR   CIN2HX
00381  0D73 16              TAB
00382  0D74 FE A04A          LDX   BF4HEX
00383  0D77 A6 00   SEARCH LDA A  0,X
00384  0D79 11              CBA
00385  0D7A 27 08            BEQ   DISADD
00386  0D7C BC A05A          CPX   ENDX
00387  0D7F 27 E6            BEQ   DONE
00388  0D81 08     SERINX INX
00389  0D82 20 F3            BRA   SEARCH
00390  0D84 FF A04A DISADD STX   BF4HEX
00391  0D87 37              PSH B
00392  0D88 BD 0CAF          JSR   NLADDR
00393  0D8B 33              PUL B
00394  0D8C FE A04A          LDX   BF4HEX
00395  0D8F 20 F0            BRA   SERINX
00396                        *
00397                  *LINK
00398                        *
00399  0D91 7E 0C8D JUMP7  JMP   I3HEX4
00400                        *
00401             *SEARCH FOR 16 BIT WORD IN MEMORY
00402                        *
00403  0D94 8D FB   SERADD BSR   JUMP7     INPUT PARAMETERS
00404  0D96 FE A04E LOOPDO LDX   TEMPX1    START ADDR
00405  0D99 BC A050 LOOPAG CPX   TEMPX2    END ADDR
00406  0D9C 27 C9            BEQ   DONE      FINISHED SO EXIT
00407  0D9E A6 00            LDA A  0,X      FETCH HI BYTE
```

```
00408  0DA0 E6 01            LDA B  1,X      FETCH LO BYTE
00409  0DA2 B1 A052          CMP A  TEMPX3    COMPARE HIBYTE
00410  0DA5 26 05            BNE   SEREXC    CONTINUE
00411  0DA7 F1 A053          CMP B  TEMPX3+1  COMPARE LO BYTE
00412  0DAA 27 03            BEQ   FONDSR    FOUND ONE
00413  0DAC 08     SEREXC INX
00414  0DAD 20 EA            BRA   LOOPAG    DO AGAIN
00415  0DAF FF A04A FONDSR STX   BF4HEX    SAVE ADDRESS
00416  0DB2 08              INX
00417  0DB3 FF A04E          STX   TEMPX1    SAVE NEXT ADDRESS
00418  0DB6 BD 0CAF          JSR   NLADDR    PRINT ADDRESS WHERE FOUND
00419  0DB9 20 DB            BRA   LOOPDO    KEEP GOING
00420                        *
00421                  *DUMP MEMORY
00422                        *
00423  0DBB BD 0CA4 DUMEM  BSR   IN2HX4    FETCH ADDRESS LIMITS
00424  0DBE FE A058          LDX   STARTX    FETCH START X
00425  0DC1 FF A04A CONTIM STX   BF4HEX    SAVE ADDRESS
00426  0DC4 BD 0CAF          JSR   NLADDR    START NEW LINE AND PRINT ADDR
00427  0DC7 86 10            LDA A  #16      SET BYTE COUNT
00428  0DC9 B7 A05F          STA A  COUNTU    INTO REGISTER
00429  0DCC FE A04A DUMLOP LDX   BF4HEX    FETCH POINTER
00430  0DCF 09              DEX             DOWN ONE
00431  0DD0 BC A05A          CPX   ENDX      END YET?
00432  0DD3 27 92            BEQ   DONE      YES EXIT
00433  0DD5 08              INX             BACK UP
00434  0DD6 BD 0FF0          JSR   HPRINT    PRINT BYTE POINTED AT
00435  0DD9 FF A04A          STX   BF4HEX    SAVE POINTER
00436  0DDC 7A A05F          DEC   COUNTU    REDUCE BYTECOUNT
00437  0DDF 26 EB            BNE   DUMLOP    KEEP GOING
00438  0DE1 20 DE            BRA   CONTIM    LINE DONE, DO ANOTHER
00439                        *
00440      *CALCULATE OFFSETS AND PRINT RESULT
00441      *IF BRANCH IS OUT OF RANGE
00442      *AN 'X' WILL BE PRINTED
00444                        *
00445      0DE3      CALOFF EQU   *
00446  0DE3 BD 0CA4          JSR   IN2HX4    FETCH ADDRESS LIMITS
00447  0DE6 5F              CLR B
00448  0DE7 5A              DEC B
00449  0DE8 5A              DEC B             SET OFFSET START VALUE
00450  0DE9 FE A058          LDX   STARTX    FETCH POINTER
00451  0DEC B6 A058          LDA A  STARTX    HI BYTE
00452  0DEF B1 A05A          CMP A  ENDX      UP OR DOWN ?
00453  0DF2 22 15            BHI   DECLOP    NEGATIVE BRANCH
00454  0DF4 B6 A059          LDA A  STARTX+1  LO BYTE
00455  0DF7 B1 A05B          CMP A  ENDX+1    UP OR DOWN ?
00456  0DFA 22 0D            BHI   DECLOP    NEGATIVE BRANCH
00457  0DFC BC A05A CNLOP  CPX   ENDX      DONE YET?
00458  0DFF 27 15            BEQ   DONCAL    YES EXIT
00459  0E01 08              INX             INCREMENT POINTER
00460  0E02 5C              INC B             INCREMENT VALUE
00461  0E03 C1 80            CMP B  #$80     OUT OF RANGE?
00462  0E05 27 16            BEQ   OUTRAN    YES
00463  0E07 20 F3            BRA   CNLOP     KEEP GOING
00464  0E09 BC A05A DECLOP CPX   ENDX      DONE YET?
00465  0E0C 27 08            BEQ   DONCAL    YES EXIT
00466  0E0E 09              DEX             DECREMENT POINTER
00467  0E0F 5A              DEC B             DECREMENT COUNT
00468  0E10 C1 7F            CMP B  #$7F     OUT OF RANGE?
00469  0E12 27 09            BEQ   OUTRAN    YES EXIT
00470  0E14 20 F3            BRA   DECLOP    KEEP GOING
00471  0E16 BD 0FF2 DONCAL JSR   SPACE     PRINT A SPACE
00472  0E19 17              TBA             TRANSFER VALUE
00473  0E1A 7E 0F46          JMP   P2HEXA    PRINT IT AND EXIT
00474  0E1D BD 0FF2 OUTRAN JSR   SPACE
00475  0E20 86 58            LDA A  #'X      SET ASCII
00476  0E22 7E 0FFA          JMP   CPRINT    PRINT IT AND EXIT
00477                        *
00478                  *TEST MEMORY
00479                        *
00480  0E25 BD 0CA4 TSTMEM BSR   IN2HX4    FETCH START AND END
00481  0E28 BD 0D57          JSR   CLRMEM+2  CLEAR MEMORY FIRST
00482  0E2B FE A058 AGAIN  LDX   STARTX
00483  0E2E A6 00   TEST   LDA A  X         FETCH DATA
00484  0E30 4C     TEST1  INC A             INCREMENT A
00485  0E31 A7 00            STA A  X
00486  0E33 B7 A060          STA A  BUFFER
00487  0E36 E6 00            LDA B  X
00488  0E38 F7 A061          STA B  BUFFER+1  SAVE FETCHED VALUE
00489  0E3B 11              CBA
00490  0E3C 26 08            BNE   THERR
00491  0E3E BC A05A          CPX   ENDX
00492  0E41 27 1A            BEQ   EXIT
00493  0E43 08              INX
00494  0E44 20 E8            BRA   TEST
00495  0E46 FF A04A THERR  STX   BF4HEX
00496  0E49 BD 0B9E          JSR   PERROR
00497  0E4C CE A060          LDX   #BUFFER   SET X TO VALUES
00498  0E4F BD 0FF0          JSR   HPRINT    PRINT FIRST TWO
00499  0E52 BD 0FF0          JSR   HPRINT    AND SECOND TWO
00500  0E55 FE A04A          LDX   BF4HEX    RESTORE X
00501  0E58 B6 A060          LDA A  BUFFER    RESTORE DATA
00502  0E5B 20 D3            BRA   TEST1
00503  0E5D 81 FF   EXIT   CMP A  #$FF     ALL PATTERNS YET?
00504  0E5F 26 CA            BNE   AGAIN     NO DO AGAIN
00505  0E61 39              RTS
00506                        *
00507                  *CONVERT ASCII TO HEX
00508                        *
00509      0E62      CONASC EQU   *
00510  0E62 BD 0C7A          JSR   CONTIN
00511  0E65 BD 0FFD          JSR   CINPUT
00512  0E68 36     CON1   PSH A
00513  0E6C BD 0FF2          JSR   SPACE
00514  0E6C 32              PUL A
00515  0E6D 7E 0F46          JMP   P2HEXA
00516                        *
00517             *OFFSET LOADER ROUTINE
00518             *IN FORMAT LT, DDDD
00519             *WHERE D IS DESTINATION
00520                        *
00521      0E70      OFLOAD EQU   *
00522  0E70 BD 0C89          JSR   CIN4HX    FETCH ACTUAL STORAGE ADDRESS
00523  0E73 FF A04E          STX   TEMPX1    SET POINTER
00524  0E76 86 11            LDA A  #$11
00525  0E78 8D 5B            BSR   PRINT     TURN ON READER COMMAND
00526  0E7A 86 3C            LDA A  #$3C
00527  0E7C B7 8007          STA A  RDRCON    TURN ON RELAY
00528  0E7F BD 0FFD OLIN   JSR   CINPUT    FETCH A CHARACTER
00529  0E82 81 53            CMP A  #'S      START?
00530  0E84 26 F9            BNE   OLIN      NOT YET
00531  0E86 BD 0FFD          JSR   CINPUT    ANOTHER
00532  0E89 81 31            CMP A  #'1
```

```
00533 0E8B 27 11          BEQ    LOAD     OK START
00534 0E8D 81 39          CMP A  #'9      END?
00535 0E8F 26 EE          BNE    OLIN     NO KEEP LOOKI
00536 0E91 86 34    OUT   LDA A  #$34     TURN OFF READER
00537 0E93 B7 8007        STA    RDRCON   TURN OFF READER
00538 0E96 86 13          LDA A  #$13
00539 0E98 8D 38          BSR    PRINT    READER OFF COMMAND
00540 0E9A 39             RTS             EXIT
00541 0E9B 7E 0C52 INHEX2 JMP    IN2HEX
00542 0E9E 7F A05E LOAD   CLR    CHKSUM   RESET CHECKSUM
00543 0EA1 8D F8          BSR    INHEX2   FETCH CHARACTER COUNT
00544 0EA3 80 02          SUB A  #2       SUBTRACT TWO
00545 0EA5 B7 A05D        STA A  BYTCNT   SET BYTE COUNT
00546 0EA8 BD 0C67        JSR    IN4HEX   FETCH ADDRESS
00547 0EAB FE A04E        LDX    TEMPX1   USE NEW OFFSET ADDRESS
00548 0EAE 8D EB   LOADST BSR    INHEX2   FETCH A DATA BYTE
00549 0EB0 BD 0B98        JSR    DANGER   BE CAREFUL
00550 0EB3 7A A05D        DEC    BYTCNT   REDUCE BYTECOUNT
00551 0EB6 27 09          BEQ    CHECK    IF LINE IS FULL
00552 0EB8 A7 00          STA A  X        NO JUST PUT IN MEMORY
00553 0EBA A1 00          CMP A  X        IS IT THERE?
00554 0EBC 26 0B          BNE    ABORT    NO MUST BE ROM
00555 0EBE 08             INX
00556 0EBF 20 ED          BRA    LOADST   OK KEEP STORING
00557 0EC1 FF A04E CHECK  STX    TEMPX1   SAVE STORAGE POINTER
00558 0EC4 7C A05E        INC    CHKSUM   CHECKSUM OK?
00559 0EC7 27 B6          BEQ    OLIN     YES IT IS
00560 0EC9 86 3F   ABORT  LDA A  #$3F
00561 0ECB 8D 05          BSR    PRINT    PRINT "?"
00562 0ECD 8D C2          BSR    OUT
00563 0ECF 7E 0B9E        JMP    PERROR
00564 0ED2 7E 0FFA PRINT  JMP    CPRINT
00565             *
00566             *OBJECT CODE DUMP ROUTINE
00567             *
00568 0ED5 BD 0CA4 PNTAB  JSR    IN2HX4   FETCH ADDRESS PARAMETERS
00569 0ED8 86 12          LDA A  #$12     SET DATA
00570 0EDA 8D F6          BSR    PRINT    START PUNCH
00571 0EDC BD 0BD5        JSR    P64NUL   FEED OUT LEADER
00572 0EDF FE A058        LDX    STARTX   FETCH ADDRESS
00573 0EE2 FF A04C        STX    TEMPX    SAVE IT
00574 0EE5 B6 A05B DUM1   LDA A  ENDX+1
00575 0EE8 B0 A04D        SUB A  TEMPX+1
00576 0EEB F6 A05A        LDA B  ENDX
00577 0EEE F2 A04C        SBC B  TEMPX
00578 0EF1 26 04          BNE    DUM2
00579 0EF3 81 20          CMP A  #32      32 BYTES PER RECORD
00580 0EF5 25 02          BCS    DUM3
00581 0EF7 86 1F   DUM2   LDA A  #31
00582 0EF9 8B 04   DUM3   ADD A  #4
00583 0EFB B7 A05C        STA A  FRMCNT   SET FRAME COUNT
00584 0EFE 80 03          SUB A  #3
00585 0F00 B7 A05D        STA A  BYTCNT   SET BYTE COUNT
00586 0F03 BD 0B65        JSR    NEWLIN   START NEW LINE
00587 0F06 CE 0F61        LDX    #TPSTRG  POINT AT TAPE STRING
00588 0F09 BD 0FCC        JSR    PDATA    PRINT THE STRING
00589 0F0C 7F A05E        CLR    CHKSUM
00590 0F0F CE A05C        LDX    #FRMCNT
00591 0F12 8D 39          BSR    OUT2H    PRINT FRAMECOUNT
00592 0F14 CE A04C        LDX    #TEMPX   FETCH POINTER
00593 0F17 8D 44          BSR    OUT4HX   PRINT ADDRESS
00594 0F19 FE A04C        LDX    TEMPX    SET POINTER
00595 0F1C 8D 2F   DUM4   BSR    OUT2H    PRINT THE DATA
00596 0F1E 7A A05D        DEC    BYTCNT   REDUCE BYTE COUNT
00597 0F21 26 F9          BNE    DUM4     KEEP DUMPING
00598 0F23 FF A04C        STX    TEMPX    SAVE ADDRESS POINTER
00599 0F26 73 A05E        COM    CHKSUM   INVERT
00600 0F29 CE A05E        LDX    #CHKSUM  SET POINTER
00601 0F2C 8D 1F          BSR    OUT2H    PRINT CHECKSUM
00602 0F2E FE A04C        LDX    TEMPX    FETCH ADDRESS
00603 0F31 09             DEX             BACK ONE
00604 0F32 BC A05A        CPX    ENDX     WAS IT THE END?
00605 0F35 26 AE          BNE    DUM1     NO KEEP GOING
00606 0F37 86 53          LDA A  #'S      SET AN S
00607 0F39 8D 97          BSR    PRINT    PRINT IT
00608 0F3B 86 39          LDA A  #'9      AND A NINE
00609 0F3D 8D 93          BSR    PRINT    PRINT ALSO
00610 0F3F BD 0BD5        JSR    P64NUL   FEED OUT TRAILER AND RTS
00611 0F42 86 14          LDA A  #$14     SET DATA
00612 0F44 20 8C          BRA    PRINT    TURN OFF PUNCH
00613             *
00614             *OUTPUT TWO HEX FROM DATA IN A
00615             *X REGISTER ALTERED
00616             *
00618 0F46 36    P2HEXA   PSH A           SAVE A DATA
00619 0F47 30             TSX             POINT AT DATA
00620 0F48 BD 0FE3        JSR    P2HXDA   PRINT IT
00621 0F4B 32             PUL A           RESTORE STACK
00622 0F4C 39             RTS             EXIT
00623             *
00624             *OUTPUT TWO HEX CHARACTERS
00625             *FROM DATA POINTED AT BY X
00626             *CHECKSUM IS UPDATED
00627             *X IS INCREMENTED ONCE
00628             *
00629 0F4D 37    OUT2H    PSH B           SAVE B
00630 0F4E BD 0FE3        JSR    P2HXDA   PRINT THE DATA
00631 0F51 09             DEX             BACK ONE ADDRESS
00632 0F52 E6 00          LDA B  0,X
00633 0F54 08             INX
00634 0F55 FB A05E        ADD B  CHKSUM
00635 0F58 F7 A05E        STA B  CHKSUM   RESTORE CHECKSUM
00636 0F5B 33             PUL B           RESTORE B REG
00637 0F5C 39             RTS
00638             *
00639             *OUTPUT FOUR HEX CHARACTERS
00640             *FROM ADDRESS POINTED AT BY X
00641             *CHECKSUM UPDATED ACCORDINGLY
00642             *X IS INCREMENTED TWICE
00643             *
00644 0F5D 8D EE  OUT4HX  BSR    OUT2H    DO FIRST BYTE
00645 0F5F 20 EC          BRA    OUT2H    DO THE SECOND AND EXIT
00646             *
00647             *TAPE FORMAT STRING
00648             *
00649 0F61 53    TPSTRG   FCB    'S,'1,4
00650             *
00651             *BNPF TAPE PUNCH ROUTINE FOR 8 BIT PROMS
00652             *FORMAT SUITABLE FOR MOST MDS READERS
00653             *SELECT PARAMETERS IN FORM PB,SSSS,EEEE
00654             *WHERE S IS START ADDRESS AN E IS END ADDRESS
00655             *INCLUSIVE. LEADERS AND TRAILERS ARE
00656             *WRITTEN TO TAPE AND THE TAPE IS PUNCHED
00657             *WITH 8 1/2 INCH FOLD MARKS

00658             *
00659 0F64 BD 0CA4 PUNBNF JSR    IN2HX4   FETCH PARAMETERS
00660 0F67 FE A05A        LDX    ENDX     FETCH END
00661 0F6A 08             INX             AND ADD ONE
00662 0F6B FF A04E        STX    TEMPX1   TO IT AND STORE
00663 0F6E 86 12          LDA A  #$12     SET DATA
00664 0F70 BD 0FFA        JSR    CPRINT   TO TURN PUNCH ON
00665 0F73 BD 0BD5        JSR    P64NUL   DO NULL LEADER
00666 0F76 8D 2C          BSR    RUBOUT   DO RUBOUTS
00667 0F78 20 03          BRA    BYTE2    SKIP NEW LINE
00668 0F7A BD 0B65 BYTE1  JSR    NEWLIN   START NEW LINE
00669 0F7D C6 08   BYTE2  LDA B  #8       SET BYTE COUNT PER LINE
00670 0F7F 86 42   BYTE3  LDA A  #'B      SET ASCII
00671 0F81 8D 77          BSR    CPRINT   PRINT A B
00672 0F83 FE A058        LDX    STARTX   FETCH POINTER
00673 0F86 BC A05A        CPX    ENDX     END YET?
00674 0F89 27 11          BEQ    ENDPUN   YES IT IS
00675 0F8B A6 00          LDA A  0,X      FETCH THE DATA
00676 0F8D 08             INX             INCREMENT POINTER
00677 0F8E FF A058        STX    STARTX   SAVE X AGAIN
00678 0F91 8D 1B          BSR    BITLOP   PRINT A BYTE
00679 0F93 86 46          LDA A  #'F      SET ASCII
00680 0F95 8D 63          BSR    CPRINT   PRINT AN F
00681 0F97 5A             DEC B           DECREMENT BYTE COUNT
00682 0F98 26 E5          BNE    BYTE3    KEEP GOING
00683 0F9A 20 DE          BRA    BYTE1    LINE DONE DO ANOTHER
00684 0F9C C6 64   ENDPUN LDA B  #100     SET COUNT
00685 0F9E 8D 06          BSR    RUB1     DO RUBOUTS
00686 0FA0 86 14          LDA A  #$14     SET CONTROL
00687 0FA2 20 56          BRA    CPRINT   TURN OFF PUNCH AND EXIT
00688 0FA4 C6 4D   RUBOUT LDA B  #77      SET COUNT
00689 0FA6 86 FF   RUB1   LDA A  #$FF     SET ALL ONES
00690 0FA8 8D 50          BSR    CPRINT   PRINT A RUBOUT
00691 0FAA 5A             DEC B           REDUCE COUNT
00692 0FAB 26 F9          BNE    RUB1     AND KEEP GOING TILL DONE
00693 0FAD 39             RTS             ITS DONE
00694             *
00695             *8 BIT PRINTER ROUTINE
00696             *
00697 0FAE 37    BITLOP   PSH B           SAVE B REGISTER
00698 0FAF C6 08          LDA B  #8       SET BIT LOOP COUNT
00699 0FB1 36    PUNB1    PSH A           PUSH ON STACK
00700 0FB2 0C             CLC             CLEAR CARRY
00701 0FB3 46             ROR A           SHIFT RIGHT
00702 0FB4 5A             DEC B           REDUCE COUNT
00703 0FB5 26 FA          BNE    PUNB1    NOT DONE YET
00704 0FB7 C6 08          LDA B  #8       SET BIT COUNT AGAIN
00705 0FB9 32    PUNB2    PUL A           FETCH A BYTE OFF STACK
00706 0FBA 46             ROR A           SHIFT INTO CARRY
00707 0FBB 24 06          BCC    PUNB3    NOT A ONE
00708 0FBD 86 50          LDA A  #'P      SET ASCII
00709 0FBF 8D 39          BSR    CPRINT   PRINT A P
00710 0FC1 20 04          BRA    PUNB4    SKIP
00711 0FC3 86 4E   PUNB3  LDA A  #'N      SET ASCII
00712 0FC5 8D 33          BSR    CPRINT   PRINT AN N
00713 0FC7 5A    PUNB4    DEC B           REDUCE BITCOUNT
00714 0FC8 26 EF          BNE    PUNB2    NOT DONE YET
00715 0FCA 33             PUL B           RESTORE B
00716 0FCB 39    POUT     RTS             EXIT , BYTE DONE
00717             *
00718             *PRINT AN ASCII DATA STRING
00719             *INCREMENT X ONCE EACH CHARACTER
00720             *EXIT WHEN EOT ENCOUNTERED
00721             *
00722 0FCC A6 00 PDATA    LDA A  X        FETCH THE DATA
00723 0FCE 81 04          CMP A  #4       IS IT EOT?
00724 0FD0 27 F9          BEQ    POUT     YUP
00725 0FD2 8D 26          BSR    CPRINT   PRINT THE CHARACTER
00726 0FD4 08             INX             UP ONE ADDRESS
00727 0FD5 20 F5          BRA    PDATA    DO IT AGAIN
00728             *
00729             *PRINT CHARACTER IN A
00730             *
00731 0FD7       HEXPRT   EQU    *
00732 0FD7 84 0F          AND A  #%00001111 MASK UPPER BITS
00733 0FD9 81 09          CMP A  #9       CHECK RANGE
00734 0FDB 23 02          BLS    HEX1     0 TO 9
00735 0FDD 8B 07          ADD A  #7       A TO F
00736 0FDF 8B 30   HEX1   ADD A  #%00110000 MAKE INTO ASCII
00737 0FE1 20 17          BRA    CPRINT   PRINT THE HEX
00738             *
00739             *PRINT TWO HEX CHARACTERS
00740             *FROM DATA POINTED TO BY X
00741             *
00742 0FE3       P2HXDA   EQU    *
00743 0FE3 A6 00          LDA A  X        FETCH THE DATA
00744 0FE5 47             ASR A
00745 0FE6 47             ASR A
00746 0FE7 47             ASR A
00747 0FE8 47             ASR A           DO UPPER NIBBLE
00748 0FE9 8D EC          BSR    HEXPRT   PRINT ONE HEX
00749 0FEB A6 00          LDA A  X        FETCH THE DATA AGAIN
00750 0FED 08             INX             INCREMENT ADDRESS
00751 0FEE 20 E7          BRA    HEXPRT   PRINT SECOND HEX AND EXIT
00752             *
00753             *PRINT TWO HEX PLUS SPACE
00754             *
00755 0FF0       HPRINT   EQU    *
00756 0FF0 8D F1          BSR    P2HXDA   PRINT THE BYTE
00757 0FF2 86 20   SPACE  LDA A  #$20     SET DATA
00758 0FF4 20 04          BRA    CPRINT   PRINT SPACE AND EXIT
00759             *
00760             *PRINT 4 HEX PLUS SPACE
00761             *
00762 0FF6       P4HEX5   EQU    *
00763 0FF6 8D EB          BSR    P2HXDA   FIRST BYTE
00764 0FF8 20 F6          BRA    HPRINT   SECOND BYTE AND EXIT
00765             *
00766             *OUTPUT ONE CHARACTER IN 'A'
00767             *B AND X UNALTERED
00768             *
00769             *
00770 0FFA       CPRINT   EQU    *
00771 0FFA 7E E1D1        JMP    $E1D1    OR OTHER USER VECTOR
00772             *
00773             *INPUT ONE CHARACTER FROM 'A'
00774             *B AND X UNALTERED
00775             *BIT 7 OF 'A' ACCUMULATOR
00776             *MASKED TO A 0
00777             *
00778 0FFD       CINPUT   EQU    *
00779 0FFD 7E E1AC        JMP    $E1AC    OR OTHER USER VECTOR
00780             *
00781                     END
```

A Text Editor for the

INTRODUCTION

Extended BASIC interpreters and some assembler packages come with a full text editor that makes it relatively easy to correct typing errors. On the other hand, 4-K and 8-K BASICs have only limited editing capabilities. The Processor Technology Software Package 1 (SP 1) and the Westminster Byte Shop XEK package have editing features that are similar to the smaller BASICs. This article describes a full text editor that can be patched into the SP 1 and XEK packages.

THE SP 1 AND XEK PACKAGES

Processor Technology has made available to the public the source listing of their SP 1. The monitor portion contains the usual commands to enter hexadecimal numbers into memory, dump a portion of memory to the console, move a block of memory, and branch to another program. (SP 1 was reviewed in the October 1976 issue of INTERFACE AGE.) The Byte Shop XEK assembler package is largely based on SP 1 except that it is disk oriented. It was reviewed in the June 1978 issue of INTERFACE AGE.

A LIMITED EDITOR

Both of these packages have provisions for two types of limited editing; one is character oriented, the other is line oriented. If a mistake is noticed immediately after typing it, the DEL (or RUB) key can be pressed. This prints a backarrow (or underline on some terminals) and deletes the character from the input buffer. If the output is sent to a PTCo video display module (VDM), the cursor backs up on the screen. Of course, the DEL key can be pressed repeatedly to delete more than one character.

REPLACING AN ENTIRE LINE

If an error is not noticed until after the line is completed, the entire line has to be retyped. But if the computerist is not an expert typist, another error may be made when retyping the line. It is difficult to produce error-free text with such a limited editor.

A FULL TEXT EDITOR

To make the SP 1 and XEK packages more useful, this author has written a full text editor that can be easily patched in. Briefly, the editor uses the line-replacement software present in these packages. The line to be edited is first located in the file buffer, copied to the input buffer, edited while in the input buffer, then copied back to the file buffer. The last step is easily accomplished by making the main package think that the edited line has been entered from the console.

THE EDIT COMMANDS

The edit commands are similar to those used in the extended BASIC distributed by MITS and TDL. If line 1130 is to be edited, give the command:

CUST 1130

for SP 1 or:

W 1130

for the XEK package. The file area is searched for the requested line. If no such line exists, an error message is printed, and control returns to the monitor. If the line is found, it is copied into the input buffer, and the line number is printed on the console. During the editing session, the H,L register pair is used as a buffer pointer.

Pressing the space bar will print the next character and advance the pointer one byte. Additional pressing of the space bar will successively display the entire line. However, there is a better way to view the text.

LOOK:

L Typing an L will display the remainder of the line, output a carriage return, a line feed and print the 4-digit line number to await the next command.

ADDITIONAL COMMANDS ARE:

SEARCH:

S Typing an S followed by a character contained in the line will move the cursor to this character and print the intervening characters.

nS Typing a number from 1 to 9 before the S will move the cursor to the n-th occurrence of the input character.

DELETE:

D Typing a D will delete the next character and embed it in a pair of backslash characters.

nD Typing a number from 1 to 9 ahead of the D will delete the next n characters. The group of deleted characters is embedded in backslashes.

INSERT:

I Typing an I followed by a string of characters will cause these characters to be inserted into the line at the cursor position. Error correction can be made at this time with the DEL key, just as during the initial entry of the line. A backarrow is printed for each character deleted. On a PTCo VDM screen the cursor is backed up deleting the character from the screen.

The insert mode is terminated one of two ways: by pressing the ESC key, in which case editing may continue, or by typing a carriage return in which case control returns to the monitor.

X Typing an X moves the cursor to the end of the line and starts the insert mode. This command is used to add characters to the end of a line.

REPLACE:

R The R command is a combination of delete and insert. A single R will delete the present character, embedding it in backslashes, then enter all additional characters into the input buffer until an ESC or carriage return command is given.

nR Typing a number from 1 to 9 ahead of the R will delete the next n characters before entering the edit mode.

XEK and PTCo Assemblers

By Alan R. Miller, Contributing Editor

QUIT:

Q   Typing a Q will return control to the monitor without altering the original line. This command is used whenever the edited line has become a mess, and there is a need to start over.

CARRIAGE RETURN:

A carriage return will cause the edited line to replace the original line in the file. This is the normal exit from the editor.

Improper editor input will ring the console bell. If an error is made in the repeat factor preceding the S, D, or R, type the correct value, then the S, D, or R. Only the last repeat factor will be used. Repeat factors in front of other edit commands are ignored.

A SAMPLE EDIT

The following example illustrates the use of the editor. Line 1 defines a file named TEST starting at address 8000 HEX. Line 2 is the system response to line 1. Lines 3 and 4 were entered from the keyboard into the file named TEST. The text editor is invoked at line 5 for the editing of line 1010 in the file. The typing of an "L" (for look) displays the entire line (line 6). The command "SD" at line 7 moves the cursor to the first occurrence of the character "D" and prints the intervening characters. A "4D" command deletes the next four characters and embeds them in backslashes (line 7). An entry of "L" prints the remainder of the line and a second "L" (line 8) prints the revised line.

For line 9, an "SC" command moves the cursor to the letter C, and an "I12345" inserts the five numbers. An ESC and an "L" finish line 9. At line 10, a "2SD" command moves the cursor to the second occurrence of the letter D. A "2R" command deletes the next two characters, then the 1236 are inserted, followed by a DEL to delete the 6 so that a 4 can be entered. An ESC and "L" finish line 10. A second "L" displays the current version (line 11). Line 12 illustrates the use of the "X" command to add characters to the end of the line. Here the final 8 was deleted with the DEL key. An ESC and an "L" produces line 13, and a carriage return finishes the edit mode, printing the line in its final form. Control then passes to monitor (line 15).

The list command shows both lines in the file. Line 1010 is now the edited version (line 17). Line 1000 is edited next (line 18). The "L" command looks at the line (line 19). A "3ST" followed by an "RN" and an ESC replaces the letter T with an R. Additional editing produces line 23 which is then scrapped with the "Q" command. The next list command shows that the original line 1000 was unaltered by the last edit (line 16 vs. line 26).

PATCHING IN THE EDITOR

SP 1 has an extra command "CUST" which causes a branch to address E000 HEX. If your SP 1 is not in PROM, you can change the CUST command to EDIT using the monitor features of SP 1. The address to patch is F28D. Give the command:

ENTR F28D (carriage return)

then type:

45 44 49 54/

the ASCII representation of the letters EDIT. Now the command:

EDIT 1010 (carriage return)

can be given to edit line 1010. If this editor must be assembled at another location, put the new address into SP 1 at address F291 HEX.

Patching XEK is a little more involved. All the letters A through Z are used for other commands. But since there are two different sets of cassette commands, one has been converted (the W command) for use with the editor. Patch location 32F0 HEX with the address of the text editor. Then the command:

W 1010

can be used to edit line 1010.

THE ASSEMBLY LISTING

The editor shown in the assembly listing is assembled for use with the XEK package. The only special patch is the equate CONIN which must be the console input routine. This is necessary since XEK won't accept the ESC command. The editor can be reassembled for SP 1 by changing the equate XEKVER from TRUE to FALSE.

After writing this text editor, the small size may be a surprise (just over 300 bytes). One wonders why text editors are not put into more programs. Editors for assemblers are simpler than those for BASIC. This is because most BASICs convert reserved words such as FOR, NEXT and GOTO to single bytes. A BASIC text editor has to first unpack each line, converting it back to the original. Then after the line is edited, it must be repacked before being placed back into the file buffer.□

```
SAMPLE RUN

> "TEST" 8000
TEST  8000  8000

1000 ;DEMONSTRATION OF TEXT EDITOR
1010 ;ABCDEFGABCDEFGABCDEFG
W 1010
1010 ;ABCDEFGABCDEFGABCDEFG
1010 ;ABC\DEFG\ABCDEFGABCDEFG
1010 ;ABCABCDEFGABCDEFG
1010 ;AB12345CABCDEFGABCDEFG
1010 ;AB12345CABCDEFGABC\DE\123←4FG
1010 ;AB12345CABCDEFGABC1234FG
1010 ;AB12345CABCDEFGABC1234FG5678←
1010 ;AB12345CABCDEFGABC1234FG567
1010 ;AB12345CABCDEFGABC1234FG567
P
1000 ;DEMONSTRATION OF TEXT EDITOR
1010 ;AB12345CABCDEFGABC1234FG567

W 1000
1000 ;DEMONSTRATION OF TEXT EDITOR
1000 ;DEMONSTRATION OF \T\NEXT EDITOR
1000 ;DEMONSTRATION OF NEXT EDITOR ADD TO END
1000 ;DEMONSTRATION OF \NEXT \EDITOR ADD TO END
1000 ;DEMONSTRATION OF EDITOR ADD TO END
1000
P
1000 ;DEMONSTRATION OF TEXT EDITOR
1010 ;AB12345CABCDEFGABC1234FG567
```

PROGRAM LISTING

```
                ; TEXTEDIT:  A TEXT EDITOR FOR THE XEX ASSEMBLER
                ;            PACKAGE AND THE PTCO SP 1
                ;
                ;           (FOR AN 8080 OR Z80 MICROPROCESSOR)
                ;
                ; PROGRAMMED BY ALAN R. MILLER
                ; NEW MEXICO TECH.  SOCORRO, NM  87801
                ;   505-835-5619              JUNE 4, 1978
                ;
0000 =          FALSE   EQU     0
FFFF =          TRUE    EQU     NOT FALSE
                ;
FFFF =          XEXVER  EQU     TRUE    ;XEX VERSION TRUE
                ;
                ; THE EDITOR COMMANDS ARE SIMILAR TO THOSE USED
                ; IN MITS AND TDL EXTENDED BASIC.
                ;
                ; FOR THE XEX VERSION, PATCH THE "V" COMMAND AT
                ; 32F0 H TO JUMP TO THE START OF THIS EDITOR.
                ; TO EDIT LINE 1010 TYPE:
                ;       V 1010
                ; AND THE PROGRAM COUNTER WILL BRANCH TO THIS EDITOR.
                ;
                ; FOR SP1, USE THE CUST COMMAND.
                ; TO EDIT LINE NUMBER 1010 TYPE:
                ;       CUST 1010
                ; AND THE PROGRAM COUNTER WILL JUMP TO E000H.
                ; IF YOUR SP 1 IS NOT IN ROM, CHANGE THE COMMAND 'CUST'
                ; TO 'EDIT' AT ADDRESS F28DH.  THE COMMAND IS THEN:
                ;       EDIT 1010
                ; THE FOUR-DIGIT LINE NUMBER WILL BE PRINTED, AND THE
                ; TEXT EDITOR WILL EXPECT ONE OF THE FOLLOWING:
                ;
                ; COMMANDS TO LOOK AT THE LINE:
                ;
                ;       EACH PRESSING OF THE SPACE BAR WILL ADVANCE THE
                ;       CURSOR ONE BYTE AND PRINT THE CORRESPONDING
                ;       CHARACTER.  OR TYPE:
                ;       L TO LOOK AT THE ENTIRE LINE.
                ;
                ; SEARCH:
                ;
                ;       S FOLLOWED BY A CHARACTER WILL MOVE THE CURSOR
                ;       TO THE INPUT CHARACTER AND PRINT THE INTERVENING
                ;       CHARACTERS.
                ;       OPTIONALLY, A NUMBER FROM 1 TO 9 MAY PRECEED THE
                ;       S IN WHICH CASE THE CURSOR WILL MOVE TO THE N-TH
                ;       OCCURANCE OF THE CHARACTER.
                ;
                ; DELETE:
                ;
                ;       D WILL DELETE THE NEXT CHARACTER OR OPTIONALLY, A
                ;       NUMBER FROM 1 TO 9 PRECEEDING THE D WILL DELETE
                ;       THE NEXT N CHARACTERS.   DELETED CHARACTERS ARE
                ;       EMBEDDED IN BACKSLASHES.
                ;
                ; INSERT:
                ;
                ;       I FOLLOWED BY ANY NUMBER OF CHARACTERS WILL INSERT
                ;       THE CHARACTERS IN THE LINE.  THE INSERT MODE IS
                ;       IS TERMINATED BY PRESSING 'ESC', IN WHICH CASE
                ;       EDITING MAY CONTINUE. OR BY A CARRIAGE RETURN IN
                ;       WHICH CASE CONTROL RETURNS TO SP 1.
                ;       WHILE IN INSERT MODE, TYPING A 'DEL'/'RUBOUT'
                ;       WILL MOVE THE CURSOR TO THE LEFT, DELETING ONE
                ;       CHARACTER EACH TIME IT IS PRESSED.
                ;       A BACKARROW IS PRINTED FOR EACH DELETE.
                ;       X WILL MOVE THE CURSOR TO THE END OF THE LINE AND
                ;       ENTER THE INSERT MODE.  THIS IS USEFUL FOR
                ;       ADDING THINGS TO THE END OF A LINE.
                ;       R (FOR REPLACE) WILL DELETE THE NEXT CHARACTER
                ;       AND ENTER THE INSERT MODE.  OPTIONALLY, TYPING
                ;       A NUMBER FROM 1 TO 9 WILL DELETE THAT MANY
                ;       CHARACTERS BEFORE ENTERING THE INSERT MODE.
                ;
                ;       Q (QUIT) IS A COP-OUT.  IF THE LINE HAS GOTTEN
                ;       AWFULLY SCREWED UP, TYPING A Q WILL RETURN THE
                ;       LINE TO ITS ORIGINAL FORM (NO QUESTIONS ASKED).
                ;
                ;       TYPE A CARRIAGE RETURN WHEN THE LINE IS IN ITS
                ;       FINAL FORM.  CONTROL WILL RETURN TO SP 1.
                ;
                ;       IMPROPER INPUT WILL RING THE KEYBOARD BELL.
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
E000            ORG     0E000H  ;START OF EDITOR
                ;
                ; EQUATES
                ;
000D =          ASCR    EQU     13      ;CARRIAGE RETURN
                ;
                        IF      XEXVER
                        TITLE   'EDITOR FOR XEX ASSEMBLER'
                ;
F808 =          CONIN   EQU     0F808H  ;CONSOLE INPUT
3057 =          CRP11   EQU     3057H   ;CR+11
3113 =          VCHK    EQU     3113H
30C6 =          IN8     EQU     30C6H
30DF =          OUT8    EQU     30DFH
30ED =          CRLF    EQU     30EDH
349D =          LINE    EQU     349DH
3036 =          EOR     EQU     3036H
36D0 =          WHAT    EQU     36D0H
3555 =          LMOV    EQU     3555H
355E =          RMOV    EQU     355EH
3526 =          FIND    EQU     3526H
32A0 =          SCRN    EQU     32A0H
40AF =          IBUF    EQU     40AFH
                        ELSE
                        TITLE   'EDITOR FOR SP 1'
                ;
                CRP11   EQU     0F047H
                VCHEX   EQU     0F293H
                IN8     EQU     0F08EH
                OUT8    EQU     0F09BH
```

```
                CONIN   EQU     IN8
                CPFL    EQU     0F0A6H
                LINE    EQU     0F476H
                EOR     EQU     0F01FH
                WHAT    EQU     0F41AH
                LMOVE   EQU     0F534H
                RMOV    EQU     0F53DH
                FIND    EQU     0F505H
                SCRN    EQU     0F213H
                IBUF    EQU     0D0ACH
                        ENDIF
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
E000 CD1331     EDIT:   CALL    VCHK    ;CHECK THAT A LINE NUMBER
                ;                               WAS ENTERED
E003 CD2635             CALL    FIND    ;FIND LINE NUMBER IN FILE
E006 C2D036             JNZ     WHAT    ;NO SUCH LINE NUMBER
E009 11AE40             LXI     D,IBUF-1
E00C 7E        EDIT2:   MOV     A,M     ;COPY FROM FILE TO IBUF
E00D FE0D               CPI     ASCR    ;CR?
E00F 12                 STAX    D
E010 CA18E0             JZ      EDIT3   ;QUIT ON CR
E013 23                 INX     H
E014 13                 INX     D
E015 C30CE0             JMP     EDIT2   ;NEXT CHAR
E018 CDED30    EDIT3:   CALL    CRLF    ;RESTART LINE
E01B 21AF40             LXI     H,IBUF
E01E 0E05               MVI     C,5
E020 46        EDIT4:   MOV     B,M     ;PRINT LINE NUMBER
E021 CDDF30             CALL    OUT8
E024 23                 INX     H
E025 0D                 DCR     C
E026 C220E0             JNZ     EDIT4
E029 0E01      EDITIN:  MVI     C,1     ;SET REPEAT TO 1
E02B CDC630    EDIN2:   CALL    IN8     ;INPUT EDIT COMMAND
E02E FE20               CPI     ' '     ;SPACE MOVES CURSOR
E030 CA79E0             JZ      EDSPC
E033 FE0D               CPI     ASCR    ;QUIT ON CR
E035 CA86E0             JZ      EDEXIT
E038 FE4C               CPI     'L'
E03A CA95E0             JZ      EDLOOK  ;LOOK AT LINE
E03D FE51               CPI     'Q'     ;QUIT
E03F CA3630             JZ      EOR     ;RESTORE OPIC LINE
E042 FE49               CPI     'I'
E044 CACAE0             JZ      EDINSR  ;INSEPT
E047 FE44               CPI     'D'
E049 CA04E1             JZ      EDDELT  ;DELETE
E04C FE53               CPI     'S'
E04E CA9BE0             JZ      EDSRCH  ;SEARCH
E051 FE52               CPI     'R'
E053 CA29E1             JZ      EDREPL  ;REPLACE
E056 FE58               CPI     'X'     ;JUMP TO END AND
E058 CA3BE1             JZ      ELEND   ;START INSERT
E05B FE57               CPI     'W'     ;SKIP TO NEXT WORD
E05D CAFFE0             JZ      EWOPD
E060 FE31               CPI     '1'     ;CHECK FOR REPEAT FACTOR
E062 DA71E0             JC      EDIN3   ;ERROR, < 1
E065 FE3A               CPI     '9'+1
E067 D271E0             JNC     EDIN3   ;ERROR, > 9
E06A DE30               SBI     30H     ;REMOVE ASCII BIAS
E06C 3C                 INR     A
E06D 4F                 MOV     C,A     ;SAVE REPEAT FACTOR
E06E C32BE0             JMP     EDIN2   ;GET LETTER COMMAND
E071 0607      EDIN3:   MVI     B,7
E073 CDDF30             CALL    OUT8    ;RING BELL ON ERROR
E076 C329E0             JMP     EDITIN  ;AND RESTART EDIT
                ;
                ; SUBROUTINE TO PRINT PRESENT CHARACTER AND
                ; ADVANCE THE POINTER
                ;
E079 46        EDSPC:   MOV     B,M     ;ADVANCE POINTER
E07A FE0D               CPI     ASCR    ;END OF LINE?
E07C CA18E0             JZ      EDIT3   ;RESTART ON END
E07F CDDF30             CALL    OUT8    ;PRINT CHAPACTER
E082 23                 INX     H
E083 C329E0             JMP     EDITIN  ;NEXT COMAND
                ;
                ; SUBROUTINE TO PRINT THE REMAINDER OF THE LINE,
                ; PLACE THE REVISED LINE IN THE FILE,
                ; AND RETURN TO SOFTWARE PACKAGE 1
                ;
E086 CDA032    EDEXIT:  CALL    SCRN    ;PRINT REST OF LINE
E089 21AF40             LXI     H,IBUF
E08C CD41E1             CALL    FCR     ;GET LINE LENGTH
E08F CD5730             CALL    CRP11   ;PUT CR AND EOF
                ;                               INDICATOR AT END OF LINE
E092 C39D34             JMP     LINE    ;REPLACE ORIGINAL LINE
                ;                               WITH EDITED LINE
                ;
                ; SUBROUTINE TO PRINT THE REMAINDER OF THE LINE
                ; AND RESTART EDIT MODE
                ;
E095 CDA032    EDLOOK:  CALL    SCRN    ;PRINT REST OF LINE AND
E098 C318E0             JMP     EDIT3   ;AND RESTART EDIT
                ;
                ; SUBROUTINE TO SEARCH FOR THE N-TH OCCURANCE
                ; OF AN INPUT CHARACTER.  ALL CHARACTERS UP TO
                ; BUT NOT INCLUDING THE SEARCH CHARACTER ARE PRINTED.
                ;
E09B CDC630    EDSRCH:  CALL    IN8     ;INPUT SEARCH CHARACTER
E09E FE20               CPI     ' '
E0A0 DA9BE0             JC      EDSRCH  ;SKIP CONTROL CHARACTER
E0A3 5F        EDSR4:   MOV     E,A     ;SAVE SEARCH BYTE
E0A4 46                 MOV     B,M
E0A5 CDDF30             CALL    OUT8
E0A8 23                 INX     H
E0A9 7E        EDSR2:   MOV     A,M     ;FETCH CURRENT BYTE
E0AA FE0D               CPI     ASCR    ;CARRIACE RETURN?
E0AC CA18E0             JZ      EDIT3   ;JUMP ON END OF LINE
E0AF BB                 CMP     E       ;COMPARE TO SEARCH BYTE
E0B0 C2B7E0             JNZ     EDSR3   ;MOVE POINTER IF NOT FOUND
E0B3 0D                 DCR     C       ;A MATCH, DECR COUNT
E0B4 CA29E0             JZ      EDITIN  ;JUMP IF N TH OCCURANCE
E0B7 47        EDSR3:   MOV     B,A
E0B8 CDDF30             CALL    OUT8    ;PRINT CURRENT BYTE
E0BB 23                 INX     H
E0BC C3A9E0             JMP     EDSR2
                ;
                ; SKIP TO NEXT WORD
```

```
                         ;
EOBF 3E20    EWORD:  MVI    A, ' '
EOC1 C3A3EO          JMP    EDSR4
                         ;
                     ; SUBROUTINE TO INSERT A CHARACTER IN THE LINE
                     ; IT MAY BE AN ESC CHARACTER
                         ;
EOC4 CD08F8  EDINSR: CALL   CONIN    ;GET INSERT CHARACTER
EOC7 FE1B            CPI    1BH      ;ESCAPE?
EOC9 CA29E0          JZ     EDITIN   ;END INSERT
EOCC FE20            CPI    ' '
EOCE DAC4ED          JC     EDINSR   ;SKIP CONTROL CHARACTERS
EOD1 FE7F            CPI    7FH      ;DELETE?
EOD3 CAF2EO          JZ     EDBACK   ;BACKUP CURSOR
EOD6 47              MOV    B, A
EOD7 CDDF30          CALL   OUT8     ;PRINT INSERTED CHAR
EODA 2B              DCX    H
EODB 7E              MOV    A, M     ;FETCH PRESENT CHAR
EODC 2F              CMA             ;COMPLEMENT IT
EODD 77              MOV    M, A     ;REPLACE IT
EODE 4F              MOV    C, A     ;USE FOR STOP ON RIGHT SHIFT
EODF CDA1E1          CALL   FCR      ;FIND CR LOCATION
EOE2 54              MOV    D, H
EOE3 5D              MOV    E, L
EOE4 23              INX    H
EOE5 CD5E35          CALL   RMOV     ;SHIFT RIGHT
EOE8 70              MOV    M, B     ;INSERT CHARACTER
EOE9 2B              DCX    H
EOEA 7E              MOV    A, M
EOEB 2F              CMA             ;COMPLEMENT BACK
EOEC 77              MOV    M, A
EOED 23              INX    H
EOEE 23              INX    H
EOEF C3C4EO          JMP    EDINSR   ;NEXT INSRT
                         ;
                     ; SUBROUTINE TO DELETE A CHARACTER (WITH DEL
                     ; COMMAND) WHEN IN INSERT MODE.  POINTER IS BACKED
                     ; UP ONE AND A BACK ARROW IS PRINTED.
                         ;
EOF2 3EAF    EDBACK: MVI    A, LOW IBUF   ;LOW HALF OF IBUF
EOF4 BD              CMP    L        ;TOO FAR BACK?
EOF5 CA29E0          JZ     EDITIN   ;YES
EOF8 2B              DCX    H
EOF9 065F            MVI    B, 5FH   ;BACK ARROW
EOFB CDDF30          CALL   OUT8     ;PRINT IT
EOFE CD2FE1          CALL   EDSHL    ;DELETE CHARACTER
E101 C3C4EO          JMP    EDINSR
                         ;
                     ; SUBROUTINE TO DELETE N CHARACTERS.  DELETED
                     ; CHARACTERS ARE IMBEDDED IN BACKSLASHES.
                         ;
E104 CDOAE1  EDDELT: CALL   EDDL2    ;DELETE AND SHIFT LEFT
E107 C329E0          JMP    EDITIN   ;NEXT COMMAND
E10A CD24E1  EDDL2:  CALL   BACKSL   ;PRINT BACKSLASH
E10D 7E              MOV    A, M     ;FETCH CHARACTER
E10E FEOD            CPI    ASCR     ;END OF LINE?
E110 C217E1          JNZ    EDDL4
E113 F1              POP    PSW      ;LINE END, RAISE STACK
E114 C318E0          JMP    EDIT3    ;RESTART LINE
E117 46      EDDL4:  MOV    B, M     ;FETCH CURRENT CHARACTER
E118 CDDF30          CALL   OUT8     ;PRINT DELETE CHAR
E11B C5              PUSH   B        ;SAVE DELETE COUNT (IN C)
E11C CD2FE1          CALL   EDSHL
E11F C1              POP    B
E120 0D              DCR    C        ;DECREMENT DELETE COUNT
E121 C217E1          JNZ    EDDL4    ;NEXT DELETE
                         ;
E124 065C    BACKSL: MVI    B, '\'   ;PRINT A BACKSLASH
E126 C3DF30          JMP    OUT8
                         ;
                     ; SUBROUTINE TO REPLACE N CHARACTERS WITH ANY
                     ; NUMBER OF OTHER CHARACTERS.  ESCAPE KEY
                     ; RETURNS TO EDIT MODE.
                         ;
E129 CDOAE1  EDREPL: CALL   EDDL2    ;DELETE AND LEFT SHIFT
E12C C3C4EO          JMP    EDINSR   ;THEN INSERT
                         ;
                     ; SUBROUTINE TO SHIFT LEFT THE REMAINDER OF THE LINE
                         ;
E12F E5      EDSHL:  PUSH   H        ;SAVE POINTER
E130 54              MOV    D, H
E131 5D              MOV    E, L
E132 13              INX    D
E133 0EOD            MVI    C, ASCR
E135 CD5535          CALL   LMOV     ;SHIFT LEFT
E138 71              MOV    M, C     ;PUT CARRIAGE RETURN IN
E139 E1              POP    H
E13A C9              RET
                         ;
                     ; SUBROUTINE TO PRINT REMAINDER OF LINE, MOVE
                     ; POINTER TO END, AND ENTER INSERT MODE
                         ;
E13B CDA032  EDEND:  CALL   SCRN     ;PRINT REMAINDER OF LINE
E13E C3C4EO          JMP    EDINSR   ;RESTART INSERT MODE
                         ;
                     ; SUBROUTINE TO FIND THE LINE LENGTH AND ADDRESS OF
                     ; THE CARRIAGE RETURN AT THE END OF THE LINE
                         ;
E141 1E01    FCP:    MVI    E, 1
E143 1C      FCR1:   INR    E
E144 7E              MOV    A, M     ;FIND CR
E145 FEOD            CPI    ASCR
E147 C8              RZ
E148 23              INX    H
E149 C343E1          JMP    FCR1
E14C                 END
```

```
000D ASCR     E124 BACKSL     F808 CONIN     30ED CPLF
3057 CRP11    EOF2 EDBACK     E104 EDDELT    E10A EDDL2
E117 EDDL4    E13B EDEND      E086 EDEXIT    E02B EDIN2
E071 EDIN3    EOC4 EDINSR     E000 EDIT      EOOC EDIT2
E018 EDIT3    E020 EDIT4      E029 EDITIN    E095 EDLOOK
E129 EDREPL   E12F EDSHL      E079 EDSPC     EOA9 EDSP2
EOB7 EDSR3    EOA3 EDSR4      E09B EDSRCH    3036 EOR
EOBF EWORD    0000 FALSE      E141 FCR       E143 FCR1
3526 FIND     40AF IBUF       30C6 IN8       349D LINE
3555 LMOV     30DF OUT8       355E RMOV      32A0 SCRN
FFFF TRUE     3113 VCHK       3EDO WHAT      FFFF XEXVEP
```

MICRO~MARKET

FIFO FLEA MARKET

FOR SALE: SWTPC 6800 system, 16K memory. CT-1024 with 16x64 mod., AC-30 cassette interface. PR-40 printer, 2 serial and 1 parallel I/O, 8K BASIC, 4K BASIC, cores. ass., games, etc. All assembled and running. $1500. Shipping paid. Write Bryan K. Miller, 1307 Wilcrest, Houston, TX 77042.

FOR SALE: Heath H8 system with 24K memory, serial & cassette interface, 2 cassette players, all standard systems software and documentation, both versions of Extended B.H. BASIC. and extra cassettes. Up and running and in excellent condition. Does not include terminal. Asking $1200 or best offer. Call Dave (201) 625-1092.

FOR SALE: 2 MITS S4 synchronous 4K memory boards, brand new, 200-300 nsec access times; PTCo ALS-8 firmware board w/8K of utility programs on PROM; Heathkit top-of-the-line IO-4510, 15MHz dual channel oscilloscope, unassembled in box; Cromemco Bytesaver, PTCo 4K Static RAM board; PAiA electronic music synthesizer, 18 modules, 3½ octave keyboard, working. Call or write Bob Stodola 1910 Beechwood Ave., St. Paul, MN 55116, (612) 698-2731.

FOR SALE: S-100 — 16x64 video interface (ASCII & block graphics), assembled $100; 8K RAM, assembled $100; Prototyping board, including buffers and regulator on board $20; Full ASCII keyboard including user-defined keys $35; Call or write Philip Klein, 1524 Sacramento St., Berkeley, CA 94702, (415) 524-9711.

FOR SALE: Compucolor computer with 8001 BASIC, dual disk drive, and 16K RAM. Used for sales demonstrations, will sell for highest offer. Video Midwest, 2212 Ingersoll Ave., Des Moines, IA 50312, (515) 244-1447.

TRADE TRS-80 PROGRAMS: Have about 50 games, graphics, etc. Send your programs on a cassette and I will record all I can on your cassette and return it to you. Level 1 only. Jim Clayton, 3520 SE Vineyard Road, Milwaukie, OR 97222.

FOR SALE: ASR 33 Teletype with stand, box of ribbons, papertape punch and reader; good condition, $600.00. Allen Tanner, 526 Cleveland Ave., Salt Lake City, UT 84105.

FOR SALE: Heath Microprocessor Course and Heath ET-3400 Microprocessor Trainer (completely assembled), including all accessories, instructions, course examination and subscription to Heath Users Group, $240.00. C. Brandt, 903 Rose St., Barnwell, SC 29812, (803) 259-7211.

FOR SALE: PerSci full size 8-inch dual disks in slimline cabinet with power supply and 1070 intelligent controller. Also CP/M Microsoft BASIC and TDL BASIC, text editor and word processor, asking $2650. First certified check, I pay UPS delivery. (Adapters available for Heathkit, S-100 & Digital Group bus.) One D.G. adapter with Zappel monitor in ROM & 2 RS232 ports, asking $210, same terms. Also, Digital Group 9" monitor in dress cabinet, $185. DG keyboard w/number cluster in dress cabinet, $175. Three DG 8K boards, $180 each. Dr. McCall, (804) 838-1950 weekdays.

FOR SALE: S-100 floppy system & 16K RAM. New 'Discus I' 8" Shugart drive system and controller, $885. 16K static, 450 ns RAM board, $290. Both from Morrow's 'Thinker Toys'. George Markle, (415) 969-4969.

FOR SALE: Heath H-8 software. 127 of the most popular computer games for $39. They run on Extended BASIC version #10.02. and require 16K of memory. Complete descriptions of games included. On high quality cassettes. Send check or money order to Mike Sexton, 37793 Colfax Court, Fremont, CA 94536.

FOR SALE: Sphere boards, CPU/2, CRT/1A. Best offer. Richars Likwartz, 827 West St., Rock Springs, WY 82901, (307) 362-5316.

FOR SALE: DEC PDP-8F minicomputer system, 16K, 2 disk drives, video terminal, ASR 33, much software. Make offer. John Robinson, 725 Berry Ln., Lexington, KY 40502, (606) 266-1509.

FOR SALE: TRS-80 software programs. Graphics, games, Psychiatrist, Biorhythm or programs written to your specifications. For complete catalog send SASE to Mike Sobetzko, 6631 Nagle Ave., Van Nuys, CA 91401.

PET USER GROUP for people interested in the Commodore PET 2001 computer. Share and exchange applications, programs and hardware expansion techniques. First year membership $5 and includes 6 issues of PET User Notes. Write Gene Beals, P.O. Box 371; Montgomeryville, PA 18936.

FOR SALE: Imsai PCS-80/35 with TV monitor, dual minidisks, 32K, IMDOS (CP/M), CBASIC, keyboard, VIO-C, and spare port for printer. Almost new; $3,000 including 6 spare diskettes. Also VIO-B for $300, 8" diskette, CBASIC and FORTRAN for $80 each, and extra 32K for $500. Ted Aho, 3965 Munkers St. SE, Salem, OR 97301, (581-9566.

FREE: Data cable with the sale of a PerSci 1070 intelligent disk controller, $500. A&T. TDL SMB board (dealer demo). 2 serial I/O's, 1 parallel, cassette interface and TDL 2K Zapple monitor in ROM and 2K RAM. Checked out, A&T $220. TDL software package "A" w/12K BASIC, Z-80 assembler, Z-TEL, Text Output Processor, all to run under CP/M format. 8" diskette, manuals and notebook, $189. Call or write Ted Nakamura, 3421 Onyx St., Torrance, CA 90503, (213) 371-8138.

WANTED: North Star BASIC Compiler. Roger T. Scaggs, 2353 Claridge Circle, Plano, TX 75075, (214) 596-1212.

FOR SALE: S-100 system. 8080 CPU w/48K of static RAM and 8K of ROM. Computer has front panel. With 108-key stand alone keyboard, video display, 15" IBM Selectric printer w/extra platen, type balls, etc. Also includes paper tape reader, Tarbell cassette system, EPROM programmer, UV eraser and over 100 programs. Reliable and nicely packaged to be sold as a system only for $3600. Jim Baumgardt, 6821 San Alto, Buena Park, CA 90620, (714) 826-7056.

WANTED: Information on users group for Intercolor 8051. Also, interested in interfacing experience with this computer via 8-bit I/O and especially software exchange. John G. Peddie, 2930 Lakeshore Ave., Oakland, CA 94610.

FOR SALE: Floppy tape peripheral. Complete with control board, 8 tapes, new and used. Uses stereo 8-track cartridge. Each cartridge can hold one program per track or 8 per cartridge. Program length limited only by tape length. The used tapes have 5 or 6 programs on each and are included free. Ready to run, including Star Trek, Othello, other games and technical programs. Will interface w/most microcomputers. Schematic incl. Will ship UPS, $100. R. Mendelson, 27 Somerset Pl., Murray Hill, NJ 07874.

FOR SALE: Digital Group TVC-64 w/documentation & OP—system cassette, Radio Shack keyboard w/Return key added. Both in one case, $175. Wanted: SWTPC CT-1024 or CT-64. Bob Howarth, Jr., RFD #1, Box 36, Lisbon, NH 03585.

FOR SALE: Apple II software-cassette; Author-Title Index program for books, records, tapes, super-fast machine language sort, 16K Blackjack, graphic, paddle input, sound; full L.V. rules, autoplay by computer, $10 ea. incl. documentation. Both $15. George Lee, 18803 S. Christina Ave., Cerritos, CA 90701.

FOR SALE: Routines for 8080 interpreter will automatically copy old BASIC programs so they can be read by the new interpreter. Source listing and instructions, $5.00. Disassembler for North Star (origins at 0000H and 5000H), $5.00 plus disk. D. Sellari, 616 N. Delaware Ave., Lindenhurst, NY 11757.

FOR SALE: TRS-80 Level 1 tapes and lists. Star Trek War (runs on 12K), list $7, tape $10. Biorhythm (4K), list $4.50, tape $7; Lunar Lander (4K), list $3, tape $5. Level 2 tapes: RAM test, $8; Lunar Lander, $5; Biorhythm, $7. J.R. Menzies, 1011 Neal Dr., Alexandria, VA 22308.

FOR SALE: SOL 20, 2 SOL 10's, Altair 8800A with p.t. motherboard, 24 kra's static RAM, 8 kra, 2-3 pts I/O board, Mits vectored interrupt, r, time clock board, iCOM dual floppy, ASR 33 TTY, TDL ZPU. Send offers, receive detailed list. Marsden, 608 Kelly, Silver City, NM 88061, (505) 538-5229.

FOR SALE: Enhancements and modifications to Microsoft KIM-1, nine digit BASIC, $15. Fast SAVE and LOAD; interrupt programs and CONTinue; GET command; append programs; paper tape using x-on — x-off; real time clock (0.1 sec.); etc. Send prepaid order or SASE for more details. Harvey Herman, 2512 Berkeley Pl., Greensboro, NC 27403.

FOR SALE: TRS-80 business programs. Accounts Receivable, Contractor's job cost, and Accountant's daily time billing. Run on level 1 OK. Moderate prices, all under $20 each. Jack Hatfield, 2895 Bedford Ave., Placerville, CA 95667.

WANTED: Individuals interested in purchasing Magnetic Stripe Reader (Byte, Feb. 1978, p. 182) in a cooperative effort for a lower price. Write to Chris Placak, Box 5756, Charlottesville, VA 22905.

FOR SALE: 8K byte, 250 nsec, static memories, $175. IBM Selectric type balls, $10. IBM Selectric tool kit, $25. IBM Selectric 10 pitch to 12 pitch conversion kit, $35. ACDC power supply, 5 to 9 volt, 20 amp, overvoltage protect, overcurrent protect, $80. Sunny Power Supply, 9 volt 25 amps, ± 18 volt-4 amps, — 9 volts-4 amps, $85. Stan Levine, 1802 Melville St., Ocean, NJ 07712, (201) 531-8305.

FOR SALE: ASR 33 like new with recent IC touch tone modem. Includes all manuals. $750. You ship. Also SWTPC CT-1024 w/scroll, 16 lines 64 char., upper/lower case, custom oak/formica cabinet. Lots of spare parts. $225. Julian Jetzer, 6400 Hawthorn Rd., Sheboygan, WI 53081, (414) 457-3366.

FOR SALE: DTE Diablo commercial Hytype I printer/keyboard, fully equipped; RS232 interface; contact E. Grossman, 410 Albany Post Rd., Croton, NY 10520.

FOR SALE: Three TRACOR 2314 type 30 megabyte disk drives w/all manuals, schematics and controller information. $600 ea. Taken from service. Michael Driscoll, 5309 Riverdale Rd., # 404, Riverdale, MD 20840, (301) 779-6425, after 6.

FOR SALE: Altair 8800B including 28K of static memory, 2SIO, ACT (with 8K MITS BASIC and games on tape), Newtech music board, plus iCOM micro-floppy with full software (FDOS III & DEBBI) plus an ADM-3A terminal. A complete system ready to run! $2495.00 complete. Call Eliot at (213) 450-1324 (day) or (213) 454-7690 (evenings).

FIFO FLEA MARKET is provided as a free service to our readers. When sending an ad please be brief in description and include price, name, address and/or phone number. There is a two month lag time. All ads this month were received before August 25th.

READER SERVICE CARD₁

INTERFACE AGE

October 1978 Issue
Void after Jan. 31, 1979
(Please type or print)

Name _____ Title _____

Company _____ Phone (A/C) _____

Address _____ ☐ Home ☐ Business

City _____ State _____ Country _____ Zip _____

ANSWER THE FOLLOWING BY CHECK-
ING ONLY ONE BOX PER QUESTION

1. I Am A:
A. ☐ Professional (Medical, Accounting, Law, Etc.)
B. ☐ Engineer (Electronics, Mechanical, Etc.)
C. ☐ Business (Purchasing, Management, Etc.)
D. ☐ Educator (Professor, Teacher, Assistant, Etc.)
E. ☐ Student
F. ☐ Hobbyist
G. ☐ Other

2. My status is:
A. ☐ Have all computing equipment
B. ☐ Need peripherals
C. ☐ Have CPU only
D. ☐ Have no equipment
E. ☐ Other

3. I need this information for:
A. ☐ Immediate purchase
B. ☐ Purchase 30/60 days
C. ☐ File
D. ☐ Literature Library
E. ☐ Other

4. My interest emphasis is:
A. ☐ Hardware
B. ☐ Firmware
C. ☐ Software
D. ☐ Other

5. My application is:
A. ☐ Hobby only
B. ☐ Business only
C. ☐ Hobby & Business only
D. ☐ Instruction Purposes
E. ☐ Research
F. ☐ Design & Development
G. ☐ Other

6. My primary source of "State-of-the-Art" information comes from:
A. ☐ Manufacturers
B. ☐ Exhibits & Conventions
C. ☐ Club meetings
D. ☐ Direct mail from manufacturers
E. ☐ Other

7. I prefer to buy:
A. ☐ Direct from manufacturers
B. ☐ Local computer/dealer (store)
C. ☐ Mail order
D. ☐ Club group purchase
E. ☐ Other

8. I look to INTERFACE AGE first for:
A. ☐ New product information
B. ☐ Software information
C. ☐ Tutorials
D. ☐ Hardware articles
E. ☐ Product advertising
F. ☐ Reviews

Please send information on items circled below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450

Over the past two years customer acceptance of Vector Graphic microcomputers has just grown and grown and grown. It isn't terribly surprising, really. After all, it does make life a lot easier when you have reliable cost effective equipment.

So it is with our new Vector MZ business system with both data processing and word processing available all in one trouble-free system.

That's good because:

By having one interactive stand-alone system with optional software for all your business functions including Accounts Payable, Accounts Receivable, General Ledger, Financial Statement, Inventory and more, plus...word processing for mailing labels, text editing of manuals, catalogs, etc., collection letters, contracts and proposals – you'll have big business power.

Known throughout the industry for highly reliable equipment, Vector Graphic's new business system incorporates: the Z-80 based Vector MZ micro-computer with built-in dual-disk quad-density storage enough to handle as many as 4,000 customers and 12,000 inventory items; Vector Graphic's "Mindless terminal"; and a Centronics 120 character per second printer.

Optional Diablo letter-quality printer is also available.

That's Vector Graphic – cost effective, versatile, reliable systems. When you get right down to it – nobody does it better.

☐ Send me details immediately.

Name _____

Company _____

Address _____

City/State/Zip _____

Dealer inquiries invited.

VECTOR GRAPHIC INC.

Vector Graphic Inc., 31364 Via Colinas
Westlake Village, CA 91361, (213) 991-2302

BUSINESS
Small Wonder

CIRCLE INQUIRY NO. 66